Restarted Pulay mixing for efficient and robust acceleration of fixed-point iterations

Phanisri P. Pratapa[a], Phanish Suryanarayana[a,*]

[a]*College of Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA*

Abstract

We present a variant of the restarted Pulay's Direct Inversion in the Iterative Subspace (DIIS) method for efficiently and robustly accelerating the convergence of fixed-point iterations. Specifically, we propose a simple modification of DIIS without any additional parameters, which we refer to as the r-Pulay method. We demonstrate the efficacy of r-Pulay in the context of the Jacobi iteration for solving large linear systems of equations, as well as in the Self Consistent Field (SCF) approach for Density Functional Theory (DFT) calculations. Overall, we find r-Pulay to be an attractive version of the restarted DIIS method.

Keywords: Pulay/Anderson mixing, Self Consistent Field (SCF) method, Electronic structure calculations, Linear system of equations, Jacobi iteration.

1. Introduction

Fixed-point iterations are regularly encountered in a variety of scientific applications. Of particular interest in this work is the Self-Consistent Field (SCF) method [1], a standard approach for determining the electronic ground state in ab-initio calculations like Density Functional Theory (DFT) [2, 3]. Since the computational time taken by electronic structure simulations is directly proportional to the number of SCF iterations required for convergence, there is great incentive in accelerating this process as far as possible [4]. Unfortunately, the rudimentary under-relaxed fixed-point iteration — commonly referred to as linear or simple mixing — typically converges extremely slowly, if at all. This is particularly true for large metallic systems at relatively low values of electronic temperature [5].

In view of the above discussion, a number of approaches have been proposed to accelerate the non-linear SCF fixed-point iteration. These include Pulay's Direct Inversion in the Iterative Subspace (DIIS) method [6] and its variants [7, 8], Broyden's quasi-Newton technique [9, 10] and its variations [11, 12, 13, 14], the Relaxed Constrained Algorithm (RCA) [15, 16], and a variety of preconditioning schemes [5, 17, 18, 19, 20]. Among these, Pulay's DIIS mixing scheme — based on the extrapolation method of Anderson [21] — has enjoyed considerable popularity and success due to its relative simplicity and overall performance [22]. Notably, the efficacy of Anderson's extrapolation scheme is not restricted to the SCF method alone [23], but also extends to a variety of other non-linear [24, 25, 26] and linear [21, 27] fixed-point problems. From a mathematical perspective, Pulay's technique can be considered to be a multisecant type method [4] that represents a specific variant of Broyden's approach [13]. In the linear setting, the DIIS approach bears remarkable similarity to the Generalized Minimal Residual (GMRES) method [28, 29, 30, 31].

*Corresponding Author (*phanish.suryanarayana@ce.gatech.edu*)

The DIIS method is occasionally found to stagnate when employed in self-consistent electronic structure calculations, resulting in unacceptably slow or non-convergence. In an effort to overcome this, Fang and Saad [4] proposed performing a restart whenever the ratio between the current and previous iteration's residual exceeds a prespecified value. Additionally, some ab-initio codes provide the option of a periodic restart within Pulay mixing [32, 33]. However, these restart techniques introduce another parameter within the DIIS method, thereby adding further complexity to the mixing scheme. In view of this, we are interested in a parameter-free restart strategy that not only prevents SCF iterations from stagnating, but also improves the efficiency and robustness of the DIIS method in general. To this end, we develop a variant of restarted Pulay for accelerating the convergence of fixed-point iterations. As an added bonus, the proposed approach is easily implementable within currently existing electronic structure codes.

The remainder of this paper is organized as follows. In Section 2, we present details of the DIIS scheme and the proposed restarted variant, which we refer to as the r-Pulay method. Subsequently, in Section 3, we validate the accuracy and efficacy of r-Pulay through examples arising in electronic structure calculations. We do so in the context of the classical Jacobi fixed-point iteration as well as the SCF method for DFT. Finally, we conclude in Section 4.

2. Restarted Pulay (r-Pulay) mixing

Consider the fixed-point problem

$$\mathbf{x} = \mathbf{g}(\mathbf{x})\,, \tag{1}$$

where $\mathbf{g} : \mathbb{C}^{N\times 1} \to \mathbb{C}^{N\times 1}$ represents the fixed-point mapping, with $\mathbb{C}$ denoting the set of all complex numbers. Perhaps the simplest attempt at a solution is an iteration of the form

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \beta \mathbf{f}_k\,, \tag{2}$$

where $\mathbf{f}_k = (\mathbf{g}(\mathbf{x}_k) - \mathbf{x}_k)$ designates the residual, and $\beta \in \mathbb{C}$ signifies the relaxation parameter. In the context of electronic structure calculations, such an approach is referred to as linear or simple mixing. Depending on the spectral properties of the residual's Jacobian, the above iteration can converge extremely slowly, if at all [5]. The Anderson/Pulay method [21, 6] attempts to overcome this limitation by generalizing Eqn. 2 to

$$\mathbf{x}_{k+1} = \bar{\mathbf{x}}_k + \beta \bar{\mathbf{f}}_k\,, \tag{3}$$

where $\bar{\mathbf{x}}_k$ and $\bar{\mathbf{f}}_k$ denote the normalized weighted averages of the previous $(m+1)$ iterates and residuals, respectively. Specifically,

$$\bar{\mathbf{x}}_k = \mathbf{x}_k - \sum_{j=1}^{m} \gamma_j \Delta\mathbf{x}_{k-m+j}\,, \tag{4}$$

$$\bar{\mathbf{f}}_k = \mathbf{f}_k - \sum_{j=1}^{m} \gamma_j \Delta\mathbf{f}_{k-m+j}\,, \tag{5}$$

where $\Delta\mathbf{x}_k = (\mathbf{x}_k - \mathbf{x}_{k-1})$, $\Delta\mathbf{f}_k = (\mathbf{f}_k - \mathbf{f}_{k-1})$, and the scalars $\Gamma_k = \begin{bmatrix}\gamma_1 & \gamma_2 & \dots & \gamma_m\end{bmatrix}^T \in \mathbb{C}^{m\times 1}$ are chosen so as to minimize the l_2-norm of the residual, i.e.,

$$\Gamma_k = \arg\min_{\Gamma_k} \|\bar{\mathbf{f}}_k\|^2\,. \tag{6}$$

It can be shown that the optimized Γ_k satisfy the relation [4]

$$\left(\mathbf{F}_k^T \mathbf{F}_k\right) \Gamma_k = \mathbf{F}_k^T \mathbf{f}_k \,, \tag{7}$$

where the residual history

$$\mathbf{F}_k = \left[\Delta \mathbf{f}_{k-m+1}, \quad \Delta \mathbf{f}_{k-m+2}, \quad \ldots, \quad \Delta \mathbf{f}_k\right] \in \mathbb{C}^{N \times m} \,. \tag{8}$$

Thereafter, the update formula in Eqn. 3 takes the form

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \beta \mathbf{f}_k - (\mathbf{X}_k + \beta \mathbf{F}_k)(\mathbf{F}_k^T \mathbf{F}_k)^{-1} \mathbf{F}_k^T \mathbf{f}_k \,, \tag{9}$$

where the iterate history

$$\mathbf{X}_k = \left[\Delta \mathbf{x}_{k-m+1}, \quad \Delta \mathbf{x}_{k-m+2}, \quad \ldots, \quad \Delta \mathbf{x}_k\right] \in \mathbb{C}^{N \times m} \,. \tag{10}$$

In the above representations of $\mathbf{X}_k$ and $\mathbf{F}_k$, a zero or negative subscript indicates a null vector. Altogether, the parameters within Pulay's approach are the relaxation parameter β and the mixing history size $(m+1)$.

The DIIS method described above utilizes the previous $(m+1)$ iterates for extrapolation after the starting $(m+1)$ iterations. Interestingly, while studying the performance of Anderson's extrapolation in the linear setting [27], we have discovered that introducing a specific type of periodic restart within the DIIS method significantly improves its performance. In Algorithm 1, we outline the resulting restarted Pulay mixing variant, which we refer to as the r-Pulay method. In this technique, all but the last columns of $\mathbf{X}_k$ and $\mathbf{F}_k$ are cleared every $(m+1)$ iterations. This relatively subtle modification not only significantly improves the overall efficiency of Pulay's DIIS method, but also makes it noticeably more robust, as demonstrated by the examples in the next section. It is worth noting that since the restart frequency coincides with the mixing history size, r-Pulay does not have any parameters apart from those already existing in Pulay mixing.

Algorithm 1: Restarted Pulay (r-Pulay) method

Input: $\mathbf{x}_0$, β, m, tol, $\mathbf{X}_0 = [\,]$ and $\mathbf{F}_0 = [\,]$
repeat $k = 0, 1, 2 \ldots$
  $\mathbf{f}_k = \mathbf{g}(\mathbf{x}_k) - \mathbf{x}_k$
  if $k > 0$ **then**
    if $k/(m+1) \in \mathbb{N}$ **then**
      $\mathbf{X}_k = [\Delta \mathbf{x}_k], \quad \mathbf{F}_k = [\Delta \mathbf{f}_k]$
    else
      $\mathbf{X}_k = \left[\mathbf{X}_{k-1}, \quad \Delta \mathbf{x}_k\right], \quad \mathbf{F}_k = \left[\mathbf{F}_{k-1}, \quad \Delta \mathbf{f}_k\right]$
    $\mathbf{x}_{k+1} = \mathbf{x}_k + \beta \mathbf{f}_k - (\mathbf{X}_k + \beta \mathbf{F}_k)(\mathbf{F}_k^T \mathbf{F}_k)^{-1} \mathbf{F}_k^T \mathbf{f}_k$
  else
    $\mathbf{x}_{k+1} = \mathbf{x}_k + \beta \mathbf{f}_k$
until $\|f_k\| < tol$;
Output: $\mathbf{x}_k$

In addition to this work, there have been a few previous efforts directed at incorporating restarts within the Pulay mixing scheme. Specifically, Fang and Saad [4] proposed setting $\mathbf{X}_k = [\,]$ and $\mathbf{F}_k = [\,]$ whenever $\|\mathbf{f}_k\| < r_p \|\mathbf{f}_{k+1}\|$, r_p being the restart parameter. Additionally, some ab-initio codes like SIESTA [32] and PARSEC [33] provide the option of restarting the DIIS method at periodic intervals so as to overcome stagnating SCF iterations. In particular, the restart in SIESTA involves setting $\mathbf{X}_k = [\,]$ and $\mathbf{F}_k = [\,]$, and

performing a linear mixing update in the subsequent iteration. However, unlike r-Pulay, the aforementioned restart strategies introduce an additional parameter into the DIIS method. Moreover, they do not retain the latest columns of $\mathbf{X}_k$ and $\mathbf{F}_k$, a feature found to have a significant impact on the performance. Overall, to the best of our knowledge, the proposed restart strategy is not available in currently existing electronic structure software, and has not been proposed in literature previously.

3. Results and Discussion

We now verify the efficacy and accuracy of the proposed r-Pulay mixing scheme through selected examples. In Section 3.1, we test r-Pulay's ability to accelerate the classical Jacobi fixed-point iteration for the solution of large, sparse linear systems of equations arising in electronic structure simulations. Next, in Section 3.2, we study the effectiveness of r-Pulay in speeding-up the Self Consistent Field (SCF) method for Density Functional Theory (DFT) calculations. We perform all computations on a workstation with the following configuration: Intel Xeon Processor E3-1220 v3 (Quad Core, 3.10GHz Turbo, 8MB), 16GB (2x8GB) 1600MHz DDR3 ECC UDIMM.

3.1. Linear systems of equations: Accelerating the Jacobi iteration

Consider the following non-periodic Poisson and complex-valued periodic Helmholtz equations arising in real-space DFT [34, 35, 36, 37] and orbital-free DFT [38, 39, 26] simulations:

$$\textit{Ex1:} \qquad -\frac{1}{4\pi}\nabla^2 V(\mathbf{r}) = \rho(\mathbf{r}) + b(\mathbf{r}) \quad \text{in } \Omega, \quad \Big\{ V(\mathbf{r}) = 0,\ \mathbf{r} \in \partial\Omega\,, \tag{11}$$

$$\textit{Ex2:} \quad -\frac{1}{4\pi}\nabla^2 V(\mathbf{r}) + Q\,V(\mathbf{r}) = P\,\rho^{\alpha}(\mathbf{r}) \quad \text{in } \Omega, \quad \begin{cases} V(\mathbf{r}) = V(\mathbf{r} + L\hat{\mathbf{e}}_i),\ \mathbf{r} \in \partial\Omega\,, \\ -\hat{\mathbf{e}}_i \cdot \nabla V(\mathbf{r}) = \hat{\mathbf{e}}_i \cdot \nabla V(\mathbf{r} + L\hat{\mathbf{e}}_i),\ \mathbf{r} \in \partial\Omega\,, \end{cases} \tag{12}$$

where $\Omega \in \mathbb{R}^3$ is a cubical domain of side L with boundary $\partial\Omega$ and unit vectors $\hat{\mathbf{e}}_i$ aligned along the edges. The fields $\rho(\mathbf{r})$ and $b(\mathbf{r})$ denote the electron and nuclear charge densities respectively, obtained by the superposition of the corresponding isolated atom quantities [39, 40]. The constants $\alpha = \frac{5}{6} + \frac{\sqrt{5}}{6}$, $P = 0.0296 + i\,0.0217$ and $Q = -0.1284 - i\,0.1269$.

We discretize the aforementioned partial differential equations using sixth-order accurate finite-differences. Specifically, we employ a mesh-size of $h = 0.5$ Bohr for the Poisson problem, with $\rho(\mathbf{r})$ and $b(\mathbf{r})$ corresponding to the Si_5H_{12}, $Si_{17}H_{36}$, $Si_{35}H_{36}$, $Si_{87}H_{76}$, $Si_{275}H_{172}$ and $Si_{525}H_{276}$ clusters. For the Helmholtz problem, we utilize $n_d = 45,\ 60,\ 75,\ 90,\ 120$, and 140 finite-difference nodes in each direction, with $\rho(\mathbf{r})$ corresponding to a system consisting of a vacancy in $3 \times 3 \times 3$ unit cells of Aluminum with lattice constant of 7.65 Bohr. The resulting linear systems of equations can be compactly written as

$$\mathbf{A}\mathbf{x} = \mathbf{b}\,; \quad \mathbf{A} \in \mathbb{C}^{N\times N}\,, \mathbf{x} \in \mathbb{C}^{N\times 1} \text{ and } \mathbf{b} \in \mathbb{C}^{N\times 1}\,, \tag{13}$$

where $\mathbf{A}$ is a sparse matrix that is symmetric positive-definite for the Poisson problem, and complex-symmetric for the Helmholtz problem. We solve these linear systems in the framework of the classical Jacobi iteration [41], wherein the fixed-point mapping

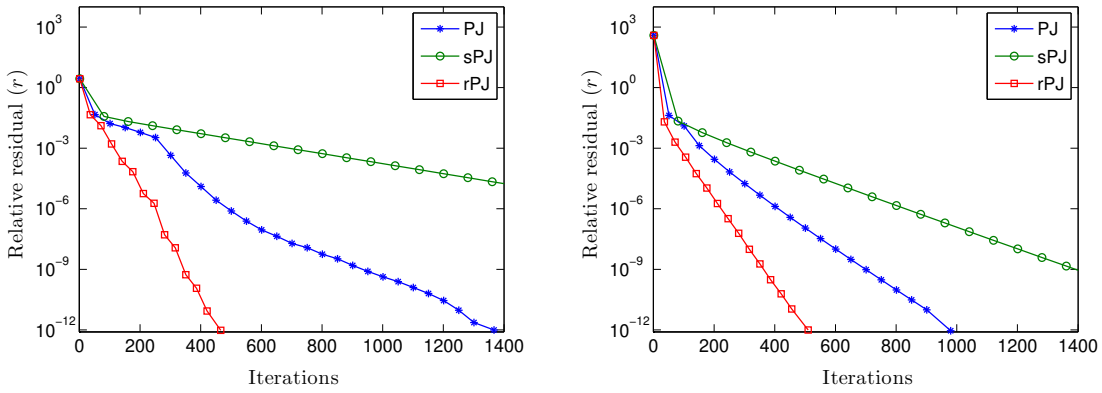
$$\mathbf{g}(\mathbf{x}) = \mathbf{D}^{-1}(\mathbf{b} - \mathbf{R}\mathbf{x})\,, \tag{14}$$

with $\mathbf{D}$ and $\mathbf{R}$ containing the diagonal and off-diagonal components of $\mathbf{A}$, respectively. We pick a vector of all ones as the starting guess $\mathbf{x}_0$, and set $tol = 1 \times 10^{-8}$ as the tolerance for convergence of the relative residual defined as

$$r_k = \frac{\|\mathbf{A}\mathbf{x}_k - \mathbf{b}\|}{\|\mathbf{b}\|} = \frac{\|\mathbf{f}_k\|}{\|\mathbf{D}^{-1}\mathbf{b}\|}\,. \tag{15}$$

In the ensuing discussion, we shall refer to the Pulay accelerated Jacobi iteration as the Pulay-Jacobi (PJ) approach, the SIESTA restarted version with restarts performed at the $(k+1)/(m+2) \in \mathbb{N}$ iterations as sPJ, and the r-Pulay variant as the rPJ method.

First, we compare the reduction of the relative residual for the PJ, sPJ and rPJ methods in Fig. 1. We select *Ex1a* (Si_5H_{12} with $h = 0.5$ Bohr), and *Ex2e* (vacancy in $3 \times 3 \times 3$ unit cells of Aluminum with $n_d = 120$) as representative examples. We choose the parameters $\{\beta, m\} = \{0.5, 3\}$, which we have found to be close to optimal for PJ in the context of the systems considered here. We observe that rPJ demonstrates significantly faster convergence than PJ, even though the chosen parameters are optimal for PJ and not rPJ. Additionally, rPJ is able to achieve extremely high accuracies while maintaining an elevated rate of convergence throughout the iteration. Indeed, rPJ's performance can be further enhanced with more judicious choice of parameters. We also note that sPJ demonstrates much slower convergence than the other two methods, which is a representative result for the linear systems considered here. In view of this, we will focus on the relative performance of PJ and rPJ for the remainder of this subsection.

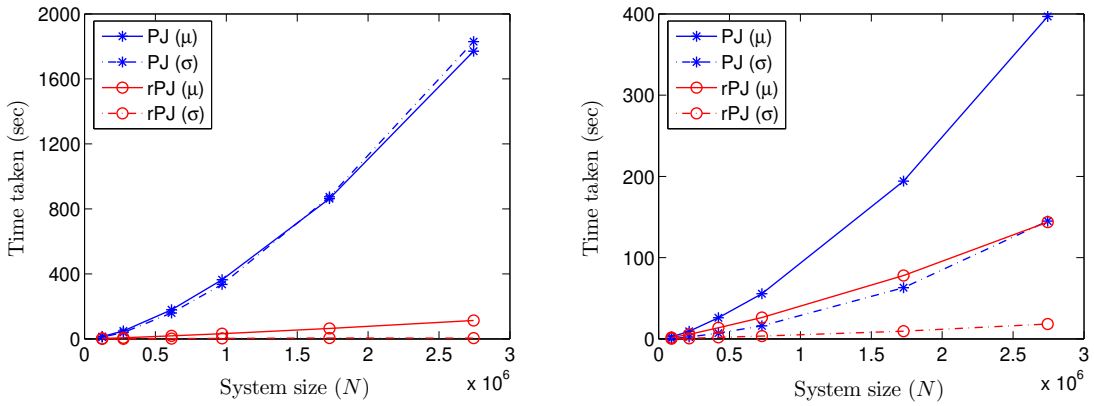


(a) *Ex1a* (b) *Ex2e*

Figure 1: Comparison of the convergence of the PJ, sPJ and rPJ methods. *Ex1a* signifies the Poisson equation for a Si_5H_{12} cluster with $h = 0.5$ Bohr. *Ex2e* denotes the Helmholtz equation for a vacancy in $3 \times 3 \times 3$ unit cells of Aluminum with $n_d = 120$.

Next, in Fig. 2, we compare the computational time taken by the PJ and rPJ methods for all the afore-described linear systems of equations. Specifically, we present the mean (μ) and standard deviation (σ) of the time taken for the parameters $\{\beta, m\} = \{0.5, 2 \text{ to } 8\}$. We observe that PJ's mean and standard deviation are noticeably larger than those of rPJ. In fact, for the biggest system in *Ex1*, PJ has larger μ and σ by factors exceeding 15 and 396, respectively. For the biggest system in *Ex2*, the corresponding ratios are close to 3 and 8, respectively. Remarkably, even though the Jacobi iteration is highly inefficient compared to Krylov subspace methods [41], rPJ is faster than the Generalized Minimal Residual Method (GMRES) [28] with a restart of 30 by factors exceeding 12 and 3 for the largest systems in *Ex1* and *Ex2*, respectively. This highlights the potential of rPJ as an efficient linear solver, particularly for large, sparse systems of equations.

Overall, we conclude that rPJ represents an accelerated and significantly more robust version of PJ. Moreover, we expect that the proposed restart strategy will also be effective in the case of non-linear fixed-point problems, particularly as the iteration proceeds towards convergence.



(a) *Ex1* (b) *Ex2*

Figure 2: Comparison of the performance of the PJ and rPJ methods. The mean and standard deviation are denoted by μ and σ, respectively. The linear systems of equations have been obtained from the discretization of the non-periodic Poisson and complex-valued periodic Helmholtz equations.

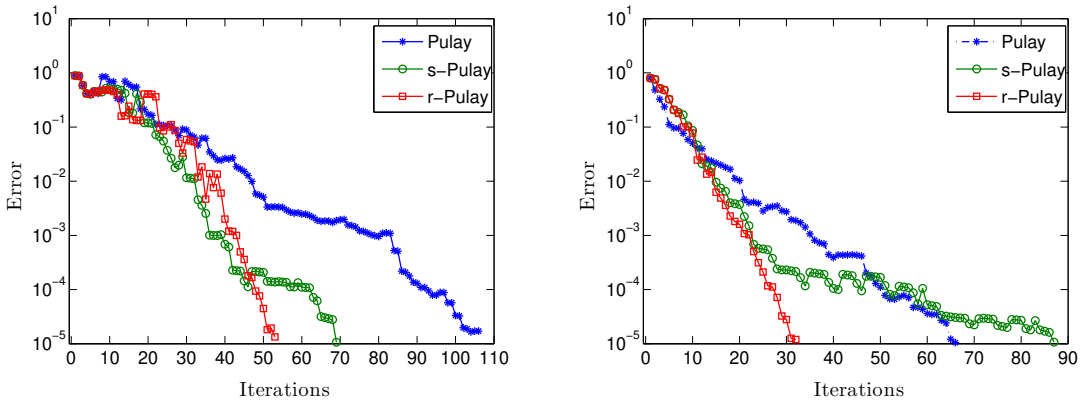
3.2. Density Functional Theory (DFT): Accelerating the Self Consistent Field (SCF) method

In this section, we study the efficacy of r-Pulay mixing in accelerating the convergence of the SCF method for DFT calculations. The SCF approach — one of the most commonly employed techniques for determining the electronic ground state in first principles calculations [23] — represents a non-linear fixed-point iteration with respect to either the electron density or the effective potential. The corresponding fixed-point mapping $\mathbf{g}(\mathbf{x})$ comprises of the electron density calculation given a Hamiltonian and effective potential evaluation given the electron density [1, 5]. Here, we perform all simulations in the framework provided by the quantum chemistry software SIESTA [42, 32]. Additionally, we denote the SIESTA variant of restarted Pulay with restarts performed at the $(k+1)/(m+2) \in \mathbb{N}$ iterations as the s-Pulay method.

In order to ensure that the results presented here are easily reproducible, we consider examples that are available as test cases within the SIESTA distribution. Specifically, we focus on the following systems: (i) *sic-slab*: 78 atom silicon carbide surface saturated by Hydrogen. (ii) *ptcda*: 2 molecules of 3, 4, 9, 10 perylenetetracarboxylic dianhydride, consisting of 76 atoms. (iii) *fe_clust_noncollinear*: 3 atom iron cluster with noncollinear spin. (iv) *batio3*: 5 atom single unit cell of barium titanate. (v) *carbon_nanoscroll*: 140 atom carbon nanoscroll saturated with Hydrogen. (vi) *si001*: 10 atom (001) Silicon surface saturated with Hydrogen. (vii) *si111-spinpol*: 22 atom (111) Silicon surface saturated with Hydrogen. In all of these examples, the only modifications made to the input files are enabling of spin polarized calculations, varying the Pulay mixing history, and using SIESTA's default tolerances for convergence of the SCF method. The motivation for including spin is that typically larger number of iterations are required for achieving convergence, which makes acceleration of the SCF process even more desirable.

We start by comparing the convergence of the Pulay, s-Pulay and r-Pulay methods during the SCF iteration. Selecting *sic-slab* and *ptcda* as representative examples, we plot the error as a function of iteration number in Fig. 3. Here, error denotes the maximum difference (in magnitude) between the density matrix

of two consecutive SCF iterations. We have employed mixing history of 5 ($m = 4$) for both *sic-slab* and *ptcda*. We observe that both r-Pulay and s-Pulay converge faster than Pulay for the *sic-slab* system. In fact, r-Pulay and s-Pulay demonstrate similar performance up to an error of around 10^{-4}, after which s-Pulay experiences a noticeable reduction in the convergence rate. The trends are similar for *ptcda*, with s-Pulay's drop in convergence rate so dramatic that it requires larger number of iterations than even Pulay to reduce the error to 10^{-5}. Altogether, r-Pulay is found to be the most efficient, and is able to achieve practical SCF tolerances in nearly half the iterations needed by Pulay mixing. This is consistent with previous results obtained in the linear setting for the Jacobi fixed-point iteration.



(a) sic-slab (b) ptcda

Figure 3: Progression of the error during the SCF iteration for the Pulay, s-Pulay and r-Pulay methods. Error denotes the maximum difference (in magnitude) between the density matrix of two consecutive SCF iterations.

Next, we compare the performance of the Pulay, s-Pulay and r-Pulay methods for the seven aforementioned electronic structure problems. Specifically, we determine the number of SCF iterations required to achieve the SIESTA default tolerances of 1×10^{-4} in the density matrix and 1×10^{-4} eV in the energy for $m = 2$, 3 and 4. Recall that $(m + 1)$ denotes the number of iterates used for extrapolation. The results so obtained are presented in Table 1. We observe that both r-Pulay and s-Pulay are significantly more efficient and robust versions of the Pulay method, with r-Pulay demonstrating the best performance overall. In particular, r-Pulay is relatively insensitive to the amount of mixing history, whereas large variations can be seen in the performance of the DIIS method. Furthermore, the proposed restart is able to speed-up Pulay mixing by up to factors exceeding 3. In fact, even for the optimal choice of $2 \leq m \leq 4$ within the Pulay method, r-Pulay demonstrates superior performance by up to factors nearing two. Intriguingly, s-Pulay consistently demonstrates superior performance to Pulay even though the opposite trend was observed in the linear setting.

System	$m=2$			$m=3$			$m=4$		
	Pulay	s-Pulay	r-Pulay	Pulay	s-Pulay	r-Pulay	Pulay	s-Pulay	r-Pulay
sic-slab	104	59	52	119	59	46	101	63	59
ptcda	93	40	30	53	34	38	59	46	28
fe_clust_noncollinear	145	79	88	93	98	61	187	160	71
batio3	146	47	39	137	38	38	61	35	34
carbon_nanoscroll	39	22	21	34	20	21	35	23	23
si001	29	24	21	26	24	22	21	22	19
si111-spinpol	41	27	21	27	24	24	21	24	20

Table 1: Number of SCF iterations taken by the Pulay, s-Pulay and r-Pulay methods to achieve the default SIESTA convergence tolerances of 1×10^{-4} in the density matrix and 1×10^{-4} eV in the energy. Number of iterates that have been utilized for mixing is $(m+1)$.

Finally, we study the statistics of the number of SCF iterations required for convergence when $2 \leq m \leq 8$. In Table 2, we present the mean (μ) and standard deviation (σ) for the Pulay, s-Pulay and r-Pulay methods. We observe that r-Pulay demonstrates the best performance among the three approaches. Specifically, r-Pulay possesses the smallest values of mean and standard deviation, further highlighting its efficiency and robustness. It is worth emphasizing that even though we have focused on spin polarized calculations, the above inferences are applicable to systems where spin is neglected. Consider for example, the systems *sic-slab* and *ptcda*. In the case of *sic-slab*, $\{\mu, \sigma\} = \{100, 43\}, \{60, 14\}$ and $\{47, 3\}$ for the Pulay, s-Pulay and r-Pulay methods, respectively. For *ptcda*, the corresponding numbers are $\{50, 28\}$, $\{30, 8\}$ and $\{26, 2\}$, respectively. It is clear that our previous conclusions are still valid.

System	Pulay		s-Pulay		r-Pulay	
	μ	σ	μ	σ	μ	σ
sic-slab	87	21	69	13	56	6
ptcda	49	22	46	9	32	4
fe_clust_noncollinear	198	108	124	34	78	25
batio3	75	48	42	11	37	3
carbon_nanoscroll	30	6	22	2	21	1
si001	23	4	22	2	20	1
si111-spinpol	25	8	23	3	21	2

Table 2: Statistics of the number of SCF iterations required for convergence when m takes values in the range of 2 to 8. The mean and standard deviation are denoted by μ and σ, respectively. The default SIESTA convergence tolerances of 1×10^{-4} in the density matrix and 1×10^{-4} eV in the energy have been employed.

As part of this work, we have performed a variety of simulations — including a number of systems not presented here — to establish the relative performance of Pulay, s-Pulay and r-Pulay. In all of these examples, we have found r-Pulay to be significantly more efficient and robust compared to Pulay. Occasionally, we have noticed the performance of s-Pulay to be slightly better than r-Pulay. As an example, for the *nanotube-c-5-0* (20 atom C(5,0) nanotube) system, $\{\mu, \sigma\} = \{21, 2\}$ and $\{24, 6\}$ for the s-Pulay and r-Pulay methods, respectively. Such results have been observed when Pulay itself requires relatively few iterations for convergence. However, for systems where convergence of the SCF is challenging, we have found that r-Pulay outperforms s-Pulay. Overall, we conclude that r-Pulay is a viable and attractive method for accelerating the SCF iteration in electronic structure calculations.

4. Concluding remarks

We have developed a variant of the restarted Pulay method (r-Pulay) — without the introduction of any additional parameters — that can be used to robustly and efficiently accelerate the convergence of fixed-point iterations. We have shown that applying the r-Pulay technique to the classical Jacobi iteration results in an extremely efficient algorithm for solving symmetric and non-symmetric linear systems of equations that is highly competitive with Krylov subspace based methods. Additionally, we have demonstrated through a number of test cases that r-Pulay efficiently and robustly accelerates the convergence of the Self Consistent Field (SCF) method in DFT calculations. In particular, r-Pulay is able to outperform the DIIS method by up to factors exceeding three. Overall, we conclude that r-Pulay represents a viable variant of the restarted DIIS method for accelerating self consistency in electronic structure calculations. This motivates further mathematical analysis into the performance of r-Pulay, making it a worthy subject for future research.

Acknowledgements

The authors gratefully acknowledge the support of National Science Foundation under Grant Number 1333500.

References

[1] J.-L. Lions, P. G. Ciarlet, Handbook of Numerical Analysis: Computational chemistry, volume 10, Gulf Professional Publishing, 2003.

[2] P. Hohenberg, W. Kohn, Physical review 136 (1964) B864.

[3] W. Kohn, L. J. Sham, Physical Review 140 (1965) A1133.

[4] H.-r. Fang, Y. Saad, Numer. Linear Algebra Appl. 16 (2009) 197–221.

[5] L. Lin, C. Yang, SIAM Journal on Scientific Computing 35 (2013) S277–S298.

[6] P. Pulay, Chemical Physics Letters 73 (1980) 393–398.

[7] D. Bowler, M. Gillan, Chemical Physics Letters 325 (2000) 473–476.

[8] K. N. Kudin, G. E. Scuseria, E. Cances, The Journal of chemical physics 116 (2002) 8255–8261.

[9] C. G. Broyden, Mathematics of computation (1965) 577–593.

[10] P. Bendt, A. Zunger, Physical Review B 26 (1982) 3114.

[11] G. Srivastava, Journal of Physics A: Mathematical and General 17 (1984) L317.

[12] D. Vanderbilt, S. G. Louie, Physical Review B 30 (1984) 6118.

[13] V. Eyert, Journal of Computational Physics 124 (1996) 271–285.

[14] L. Marks, D. Luke, Physical Review B 78 (2008) 075114.

[15] E. Cancès, C. Le Bris, International Journal of Quantum Chemistry 79 (2000) 82–90.

[16] E. Cances, C. Le Bris, ESAIM: Mathematical Modelling and Numerical Analysis 34 (2000) 749–774.

[17] G. Kerker, Physical Review B 23 (1981) 3082.

[18] K.-M. Ho, J. Ihm, J. Joannopoulos, Physical Review B 25 (1982) 4260.

[19] D. Raczkowski, A. Canning, L. Wang, Physical Review B 64 (2001) 121101.

[20] P.-M. Anglade, X. Gonze, Physical Review B 78 (2008) 045126.

[21] D. G. Anderson, Journal of the Association for Computing Machinery 12 (1965) 547–560.

[22] K. N. Kudin, G. E. Scuseria, ESAIM: Mathematical Modelling and Numerical Analysis 41 (2007) 281–296.

[23] G. Kresse, J. Furthmüller, Physical Review B 54 (1996) 11169.

[24] V. Ganine, N. Hills, B. Lapworth, International Journal for Numerical Methods in Fluids 71 (2013) 939–959.

[25] J. Willert, W. T. Taitano, D. Knoll, Journal of Computational Physics 273 (2014) 278–286.

[26] S. Ghosh, P. Suryanarayana, arXiv preprint arXiv:1412.8250 (2014).

[27] P. P. Pratapa, P. Suryanarayana, J. E. Pask, Under Review (2015).

[28] Y. Saad, M. H. Schultz, SIAM Journal on scientific and statistical computing 7 (1986) 856–869.

[29] T. Rohwedder, R. Schneider, Journal of mathematical chemistry 49 (2011) 1889–1914.

[30] H. F. Walker, P. Ni, SIAM Journal on Numerical Analysis 49 (2011) 1715–1735.

[31] F. A. Potra, H. Engler, Linear Algebra and its Applications 438 (2013) 1002–1011.

[32] E. Artacho, E. Anglada, O. Diéguez, J. D. Gale, A. García, J. Junquera, R. M. Martin, P. Ordejón, J. M. Pruneda, D. Sánchez-Portal, et al., Journal of Physics: Condensed Matter 20 (2008) 064208.

[33] L. Kronik, A. Makmal, M. L. Tiago, M. Alemany, M. Jain, X. Huang, Y. Saad, J. R. Chelikowsky, physica status solidi (b) 243 (2006) 1063–1079.

[34] J. E. Pask, P. A. Sterne, Phys. Rev. B 71 (2005) 113101.

[35] P. Suryanarayana, V. Gavini, T. Blesgen, K. Bhattacharya, M. Ortiz, Journal of the Mechanics and Physics of Solids 58 (2010) 256 – 280.

[36] P. Suryanarayana, K. Bhattacharya, M. Ortiz, Journal of Computational Physics 230 (2011) 5226 – 5238.

[37] J. E. Pask, N. Sukumar, S. E. Mousavi, International Journal for Multiscale Computational Engineering 10 (2012) 83–99.

[38] P. Motamarri, M. Iyer, J. Knap, V. Gavini, Journal of Computational Physics 231 (2012) 6596 – 6621.

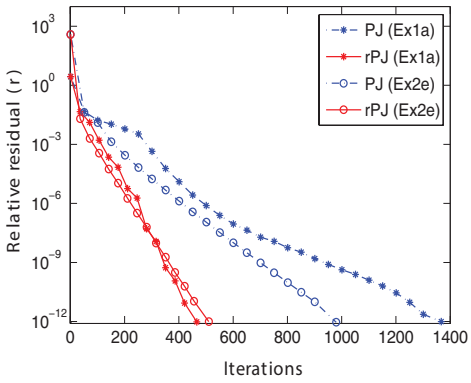
[39] P. Suryanarayana, D. Phanish, Journal of Computational Physics 275 (2014) 524 – 538.

[40] P. Suryanarayana, K. Bhattacharya, M. Ortiz, Journal of the Mechanics and Physics of Solids 61 (2013) 38–60.

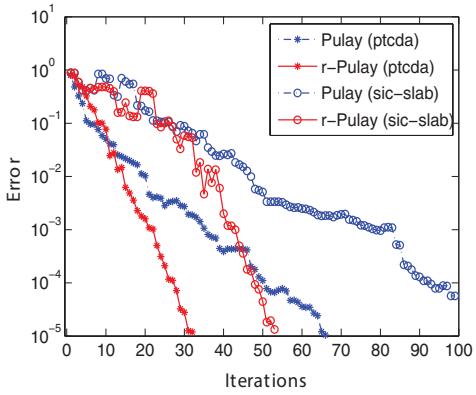
[41] Y. Saad, Iterative Methods for Sparse Linear System (2nd ed), SIAM, 2003.

[42] J. M. Soler, E. Artacho, J. D. Gale, A. García, J. Junquera, P. Ordejón, D. Sánchez-Portal, Journal of Physics: Condensed Matter 14 (2002) 2745.

Graphical Abstract



Linear systems: Jacobi iteration



DFT: SCF iteration

System	Pulay		r-Pulay	
	μ	σ	μ	σ
fe_clust_noncollinear	198	108	78	25
batio3	75	48	37	3
ptcda	49	22	32	4
sic-slab	87	21	56	6
carbon_nanoscroll	30	6	21	1
si001	23	4	20	1
si111-spinpol	25	8	21	2

DFT: Statistics of the number of SCF iterations

Comparison of the proposed r-Pulay mixing with Pulay mixing