Numerical Algorithms (2022) 90:301–321
https://doi.org/10.1007/s11075-021-01189-4

ORIGINAL PAPER

Implicit algorithms for eigenvector nonlinearities

Elias Jarlebring[1] · Parikshit Upadhyaya[1]

Received: 22 May 2020 / Accepted: 13 August 2021 / Published online: 2 September 2021

Abstract
We study and derive algorithms for nonlinear eigenvalue problems, where the system matrix depends on the eigenvector, or several eigenvectors (or their corresponding invariant subspace). The algorithms are derived from an implicit viewpoint. More precisely, we change the Newton update equation in a way that the next iterate does not only appear linearly in the update equation. Although the modifications of the update equation make the methods implicit, we show how corresponding iterates can be computed explicitly. Therefore, we can carry out steps of the implicit method using explicit procedures. In several cases, these procedures involve a solution of standard eigenvalue problems. We propose two modifications, one of the modifications leads directly to a well-established method (the self-consistent field iteration) whereas the other method is to our knowledge new and has several attractive properties. Convergence theory is provided along with several simulations which illustrate the properties of the algorithms.

Keywords Eigenvector nonlinearity · Inexact Newton · Implicit Newton · SCF

1 Introduction

Let $M \subset \mathbb{R}^{n\times n}$ denote the set of symmetric $n \times n$-matrices. Let $A : \mathbb{R}^{n\times p} \to M$, $p \leq n$. We consider the problem of finding $V \in \mathbb{R}^{n\times p}$ and a symmetric $S \in \mathbb{R}^{p\times p}$ such that

$$A(V)V = VS, \tag{1a}$$
$$V^TV = I. \tag{1b}$$

✉ Parikshit Upadhyaya
pup@kth.se

Elias Jarlebring
eliasj@kth.se

[1] Department of Mathematics, SeRC - Swedish e-Science Research Center, KTH Royal Institute of Technology, Lindstedtsvägen 25, SE-11428, Stockholm, Sweden

This is the general formulation of the eigenvector-dependent nonlinear eigenvalue problem. In our work, A satisfies $A(VP) = A(V)$ for any non-singular matrix P such that the range of V can be seen as an invariant subspace of A. This property (and a notion of invariant subspace) is characterized in Section 2.1, where we also provide a problem transformation applicable when the condition is not satisfied.

If $p = 1$, the setting reduces to a class of problems which has received considerable attention, mostly in application-specific settings, as we further discuss below. In this case, we need to determine $v \in \mathbb{R}^n$ and $\lambda \in \mathbb{R}$ such that

$$A(v)v = \lambda v \tag{2}$$

where $\|v\| = 1$.

A number of algorithms have been proposed for the above problems, both for $p = 1$ and the general case. In this paper, we propose to derive algorithms based on implicit formulations, in particular based on implicit improvements of Newton's method. One proposed algorithm leads to a linearly convergent well-established method, whereas the other approach leads to a new method with quadratic convergence. Both of the implicit approaches have advantages for certain problem classes that we characterize.

Our approach is based on viewing iterative eigenvalue solvers (for eigenvector nonlinearities) as modifications of Newton's method. This has also been done for standard eigenvalue problems, already by Wilkinson and Peters [23]. The Newton's method viewpoint of iterative methods has also been used in the derivation of algorithms for nonlinear eigenvalue problems with eigenvalue nonlinearity, e.g., [15, 21, 32]. See also the recent review paper [30] and to our knowledge the first publication in this direction by Unger [33].

One of the most important applications for (1) is within the field of quantum mechanics and electronic structure calculations. Discretization methods in combination with the Hartree-Fock approximation or the Kohn-Sham equations lead to problems of type (1). See standard literature in quantum chemistry [29]. For a survey of numerical methods, see [28]. Considerable application-specific research has been carried to specialized algorithms for this problem, mainly based on the *self-consistent field iteration* (SCF). SCF is an iterative method that involves solving a linear eigenvalue problem in each step until convergence or *self-consistency*. The convergence of SCF and its variants has been studied in a number of works which can be classified into two broad categories: the optimization-based approach of looking at (1) as the optimality conditions of a minimization problem [8, 18–20] or different matrix analysis–based approaches [34, 35]. For a discussion of similarities and differences among the two approaches, see [9]. Strategies for accelerating the convergence of SCF have also been studied well, e.g., [24, 25].

The special case $p = 1$ has very important applications in quantum physics. Characterization of the ground state of bosons is usually done with the Gross-Pitaevskii equation, see, e.g., references in [1] (and also [16]), whose spatial discretization is of the form (2). Although SCF can be used in this case too, the more common techniques involve discretization of a gradient flow. See [5], and references therein. Also applicable to the Gross-Pitaevskii equation is the result in our paper [16], which has similarities with the current approach in the sense that we both use higher order terms (represented with a specific type of Jacobian).

Another class of applications where $p = 1$ arises is in data science, for example, applications such as spectral clustering which rely on computing eigenpairs of the p-Laplacian [7, 13, 22]. See [4] for a Rayleigh quotient minimization approach for Fisher linear discriminant analysis, which is used in pattern recognition and classification. In [31], the authors propose a new model for the core-periphery detection problem in network science (in the sense of [6]) and show its equivalence to the $p = 1$ problem.

Our approach is based on flavors of Newton's method and there are many other approaches also based on Newton's method for related problems. A popular technique is based on deriving iterative methods from a geometric perspective, [12], often with optimization techniques as in [36]. Our algorithms are not derived from an optimization method, and more importantly, our implicit construction directly provides features which are not directly available from a geometry perspective, e.g., the attractive property illustrated in Section 5.3.

We note that in the context of some applications, the methods of this paper are naturally viewed as the second stage in a two-stage approach outlined as follows:

1. Obtain an approximation of the solution of interest with a method which has attractive global convergence properties.
2. Improve the solution with a fast locally convergent method.

For example, in the context of the Gross-Pitaevskii equation, the popular gradient flow methods [5] often converge to the solution of interest (the ground state) but can be slow if a small step-length is required in the time-stepping scheme. A gradient flow method can be used in the first stage, followed by a fast locally convergent method in the second stage to obtain an accurate solution. The method in the second stage can be initialized with the solution from the first stage. Some NEPv problems are easier to solve, in the sense that there exist algorithms based on a companion linearization [10], which is another example of a method to be used in the first stage.

The contributions of the paper can be summarized as follows. In Section 2.1, we introduce the concept of basis invariance. This allows us to derive an alternate characterization of (1) in terms of an associated Jacobian. We introduce our implicit algorithms in Section 3 motivated by this result. Explicit procedures to carry out these algorithms are derived and studied in Section 4. In Section 5, we provide convergence results for these algorithms and Section 6 contains numerical examples, illustrating advantages of our approach.

We will extensively use vectorization and devectorization and introduce the following shorthand. Small letters denote the vectorization of capital letters. For example,

$$x = \mathrm{vec}(X), \quad v^{(k)} = \mathrm{vec}(V^{(k)}), \quad s^{(k)} = \mathrm{vec}(S^{(k)}), \quad i_p = \mathrm{vec}(I_p).$$

For any $H : \mathbb{R}^n \to \mathbb{R}^n$, the operator $\frac{dH}{dv}$ denotes forming the Jacobian of H with respect to v, where v denotes vectors in $\mathbb{R}^n$. Also, $\left(\frac{dH(v)}{dv}\hat{v}\right)_{\hat{v}=v}$ denotes the directional derivative of H evaluated at v in the direction of v.

2 Preliminaries

2.1 Notion of invariant subspace

In order to appropriately generalize the concept of invariant pairs, we will throughout the paper make the following assumption on A.

Assumption 1 (**Basis invariance**) *We consider $A : \mathbb{R}^{n\times p} \to M$ such that it is a function of the outer product of W, i.e.,*

$$A(W) = B(WW^T) \tag{3}$$

for some $B : M \to M$. Moreover, we assume that

$$B(X) = B(h(X)) \tag{4}$$

for any $X \in M$, where $h : \mathbb{R} \to \mathbb{R}$ denotes the heaviside function

$$h(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{if } x \leq 0 \end{cases} . \tag{5}$$

Note that h is defined in a matrix function sense, for example, using diagonalization in [14, Definition 1.2]. Assumption 1 is a generalization of the scaling invariance property for the case $p = 1$ in [16]. If $p = 1$ and $v \in \mathbb{R}^n$, then for any $\alpha \in \mathbb{R}$,

$$A(\alpha v) = B(\alpha^2 vv^T) = B(h(\alpha^2 vv^T)) = B(h(vv^T)) = A(v).$$

Moreover, Assumption 1 leads to the fact that $A(W) = A(WP)$ for invertible P, as we shall illustrate in the following theorem. This is important in our context, since it allows us to interpret the columns of W as a basis of an invariant subspace, and A can be viewed as a function of a subspace, i.e., it is a function of a vector space, and independent of the basis.

Theorem 1 *If A satisfies the basis invariant conditions (3) and (4) then $A(W) = A(WP)$ for any non-singular matrix $P \in \mathbb{R}^{p\times p}$ and $W \in \mathbb{R}^{n\times p}$ where $n \geq p$ and $\operatorname{rank}(W) = p$.*

Proof Let $W = QU$ for some invertible matrix U and $Q \in \mathbb{R}^{n\times p}$ orthogonal. Let V, Λ_+ be a diagonalization of UU^T, i.e., $UU^T = V\Lambda_+V^T$, where V is orthogonal and Λ_+ is a positive diagonal matrix. Then,

$$h(WW^T) = h(QUU^TQ^T) = h(QV\Lambda_+V^TQ^T) = QVh(\Lambda_+)V^TQ^T = QQ^T.$$

This along with (3) and (4) gives us

$$A(W) = B\left(h(WW^T)\right) = B(QQ^T) = A(Q).$$

If we let $W = QR$ be a QR factorization of W, we see that $A(W) = A(Q)$ with $U = R$, and $A(WP) = A(Q)$ with $U = RP$. This shows that $A(W) = A(WP)$, which concludes the proof. □

Since S is symmetric, it can be diagonalized as $S = Q_s \Lambda_s Q_s^T$ where Q_s is orthogonal. Problem (1) can be reformulated using Theorem 1 as

$$A(V)V = VQ_s\Lambda Q_s^T \implies A(VQ_s)VQ_s = VQ_s\Lambda_s. \tag{6}$$

showing that a solution to (1) can be diagonalized.

Example 1 (Transformation to basis invariant form) The heaviside function usually does not appear directly in the standard formulation in NEPv applications, e.g., those mentioned in the introduction, but can be obtained easily. In the context of the self-consistent field iteration for a simplified version of a quantum chemistry problem, we want to solve the equation

$$H(V)V = VS \tag{7}$$

where, e.g., $H(V) = H_0 + \mathrm{diag}(VV^T)$, which does not satisfy (3) and (4). This can be transformed to a problem satisfying (3) and (4) by defining

$$A(V) := H_0 + \mathrm{diag}(h(VV^T)). \tag{8}$$

A pair (V, S) is a full rank solution to (7) if and only if it is a solution to (1) with A defined as in (8). However, the similarity transformation of a solution, i.e., $(VP, P^{-1}SP)$ is a solution to (1), but not (7).

2.2 Jacobian properties

We will denote the Jacobian as follows, and we directly characterize a theoretical property as a consequence of Assumption 1.

Definition 1 (Left-hand side Jacobian) The Jacobian of the vectorization of the LHS of (1a) is denoted as $J : \mathbb{R}^{np} \to \mathbb{R}^{np \times np}$ and given by

$$J(v) = I_p \otimes A(V) + \left(\frac{d}{dv}(I_p \otimes A(V))\hat{v} \right)_{\hat{v}=v} \tag{9}$$

In the analysis and derivation of our algorithms, we need an intermediate problem where the orthogonalization is done with respect to a constant C matrix. More precisely, we study this problem:

$$A(W)W = WZ, \tag{10a}$$
$$C^T W = I. \tag{10b}$$

Equivalence of (10) and (1) is discussed in Section 3. The vectorized form of (10) can now be written as

$$F(W, Z) := \begin{pmatrix} (I_p \otimes A(W))w - (I_p \otimes W)z \\ (I_p \otimes C^T)w - i_p \end{pmatrix} = \begin{pmatrix} (I_p \otimes A(W))w - (Z^T \otimes I_n)w \\ (I_p \otimes C^T)w - i_p \end{pmatrix} = 0. \tag{11}$$

The methods we propose will work better for problems where the Jacobian evaluated in the solution is non-singular. The Jacobian of (11) in the fixed point is given by

$$\begin{bmatrix} J(w_*) - Z_*{}^T \otimes I_n & -I_p \otimes W_* \\ I \otimes C^T & 0 \end{bmatrix}. \tag{12}$$

As a consequence of Assumption 1, we conclude the following generalization of [16, Lemma 2.1], which shows a relationship between the eigenpairs of J and A. We exploit this relationship later in Sections 3 and 4 where we formulate and derive our algorithms respectively.

Theorem 2 (**Eigenproblem equivalence**) *For any $v \in \mathbb{R}^{np}$, we have*

$$J(v)v = (I_p \otimes A(V))v.$$

Proof From (9), we have

$$\big(J(v) - (I_p \otimes A(V))\big)\, v = \left(\frac{d}{dv}(I_p \otimes A(V))\hat{v}\right)_{\hat{v}=v} v.$$

Interpreting $\left(\frac{d}{dv}(I_p \otimes A(V))\hat{v}\right)_{\hat{v}=v}$ as a directional derivative at v in the direction of v and evaluated at v, we have

$$\begin{aligned}\left(\tfrac{d}{dv}(I_p \otimes A(V))\hat{v}\right)_{\hat{v}=v} v &= \lim_{\epsilon\to 0} \frac{\big(I_p\otimes B\big((V+\epsilon V)(V+\epsilon V)^T\big) - I_p \otimes B\big(VV^T\big)\big)v}{\epsilon} \\ &= \lim_{\epsilon\to 0} \frac{\big(I_p\otimes B\big(h\big((\epsilon+1)^2 VV^T\big)\big) - I_p \otimes B\big(VV^T\big)\big)v}{\epsilon} \\ &= \lim_{\epsilon\to 0} \frac{\big(I_p\otimes B\big(VV^T\big) - I_p \otimes B\big(VV^T\big)\big)v}{\epsilon} \\ &= 0.\end{aligned}$$

This completes the proof. □

3 Implicit algorithms

3.1 Constant orthogonalization Newton's method

The basis of our derivation is a variation of the nonlinear system of (1). The property that $A(VP) = A(V)$ implies that we can view V as a basis of a subspace, and therefore we can consider an equivalent formulation with a different orthogonality condition (10) where $C^{n\times p}$ is any fixed vector (and is typically chosen as an approximation of W). The problems (10) and (1) are equivalent for any W with full column rank since if we let $W^T W = R^T R$ be a Cholesky factorization of $W^T W$ we can define $V = WR^{-1}$ such that $V^T V = I$ and $A(V) = A(W)$. This is direct consequence of the Grassman manifold description, as explained in the convergence theory (of SCF) in [3].

The constant orthogonalization formulation (10) will be used to derive an algorithm. Although (1) and (10) are equivalent, we use (10) in the analysis for simplicity.

Note that for $p > 1$, (1) has a continuum of solutions because V is an orthogonal basis of a subspace and any other orhogonal basis is also a solution (for different S). The solutions to (10) are in general isolated.

Standard Newton's method for the vectorized form (10) is

$$-F^{(k)} = \begin{bmatrix} J(w^{(k)}) - Z^{(k)^T} \otimes I_n & -I_p \otimes W^{(k)} \\ I_p \otimes C^T & 0 \end{bmatrix} \begin{bmatrix} \Delta w^{(k)} \\ \Delta z^{(k)} \end{bmatrix} \tag{13}$$

where the Δ-matrices are updates, $w^{(k+1)} = w^{(k)} + \Delta w^{(k)}$ and $z^{(k+1)} = z^{(k)} + \Delta z^{(k)}$, with where

$$F^{(k)} := F(W^{(k)}, Z^{(k)}).$$

Lemma 1 *Let $w^{(k)}$, $z^{(k)}$, $k = 1, \ldots$, be a sequence of vectors that satisfy* (13)*. Then,*

(a) $C^T W^{(k)} = I_p$ *for* $k = 2, 3, \ldots$

(b) *If the Jacobian evaluated in* (W_*, Z_*) *(given by* (12)*) is invertible, then the convergence is quadratic.*

Proof If $C^T W^{(k)} = I_p$ then the second subequation of (12) gives

$$(I_p \otimes C^T)\Delta w^{(k)} = i_p - (I_p \otimes C^T) w^{(k)} \implies C^T W^{(k+1)} = I_p,$$

which proves (a). The statement (b) is a standard result about the convergence of Newton's method □

In this work, we consider two modifications of Newton's method as formulated in (13). Both lead either to new methods (which have some attractive properties) or well-established methods suggesting that the methods can be viewed as Newton-like methods.

– We modify the (1,2) block of the Jacobian to $I_p \otimes W^{(k+1)}$.

$$-F^{(k)} = \begin{bmatrix} J(w^{(k)}) - Z^{(k)^T} \otimes I_n & -I_p \otimes W^{(k+1)} \\ I_p \otimes C^T & 0 \end{bmatrix} \begin{bmatrix} \Delta w^{(k)} \\ \Delta z^{(k)} \end{bmatrix} \tag{14}$$

This is analogous to the modification [15, Equation (1.10)] which directly leads to the method of successive linear problems [27]. With the techniques of the next section, this leads to Algorithm 1.

– We modify the (1,1) block of the Newton's method Jacobian from $J(w^{(k)})$ to $I_p \otimes A(W^k)$, in addition to the modification done to obtain Algorithm 1.

$$-F^{(k)} = \begin{bmatrix} I_p \otimes A(W^{(k)}) - Z^{(k)^T} \otimes I_n & -I_p \otimes W^{(k+1)} \\ I_p \otimes C^T & 0 \end{bmatrix} \begin{bmatrix} \Delta w^{(k)} \\ \Delta z^{(k)} \end{bmatrix} \tag{15}$$

We will show that this leads to Algorithm 2, which is the well-known SCF iteration.

4 Reformulation for direct computation

Both updates (14) and (15) correspond to implicit methods. We will illustrate several situations where we can generate iterates that satisfy the implicit algorithms update equations in an explicit way.

4.1 Algorithm 2

Although Algorithm 2 is a modification of Algorithm 1, we start our discussion with Algorithm 2 since it leads to a well-established method. We can obtain Algorithm 2 from (15) by multiplying out the first subequation of (15) as follows.

$$\begin{aligned}&\left((I_p \otimes A(W^{(k)})) - ({Z^{(k)}}^T \otimes I_n)\right)(w^{(k+1)} - w^{(k)}) - (I_p \otimes W^{(k+1)})(z^{(k+1)} - z^{(k)})\\ &= -\left((I_p \otimes A(W^{(k)})) - ({Z^{(k)}}^T \otimes I_n)\right) w^{(k)}\end{aligned}$$

Cancellation of terms leads to

$$\left(I_p \otimes A(W^{(k)})\right) w^{(k+1)} = (Z^{(k+1)} \otimes I_n) w^{(k+1)}. \tag{16}$$

Devectorizing this system gives the following result.

Theorem 3 *Suppose $C^T W^{(k)} = I_p$. Then, the pair $(W^{(k+1)}, Z^{(k+1)})$ satisfies the update (15) if and only if it satisfies*

$$A(W^{(k)})W^{(k+1)} = W^{(k+1)} Z^{(k+1)}, \tag{17}$$

and $C^T W^{(k+1)} = I_p$.

Equation (17) gives a practical way to compute the iterates from (15). Given $W^{(k)}$, we compute $(W^{(k+1)}, Z^{(k+1)})$ as an invariant pair of $A(W^{(k)})$. This is the well-known SCF algorithm.

4.2 Algorithm 1

The first subequation in (15) implies that

$$\begin{aligned}&\left(J(W^k) - ({Z^{(k)}}^T \otimes I_n)\right)(w^{(k+1)} - w^{(k)})\\ &= (I_p \otimes W^{(k+1)})(z^{(k+1)} - z^{(k)}) + ({Z^{(k)}}^T \otimes I_n) w^{(k)} - (I_p \otimes A(W^{(k)})) w^{(k)}\\ &= \left(\left({Z^{(k+1)}}^T - {Z^{(k)}}^T\right) \otimes I_n\right) w^{(k+1)} + ({Z^{(k)}}^T \otimes I_n) w^{(k)} - (I_p \otimes A(W^{(k)})) w^{(k)}.\end{aligned} \tag{18}$$

From Theorem 2, we have $J(w^{(k)})w^{(k)} = \left(I_p \otimes A(W^k)\right) w^{(k)}$. Using this in (18) leads to the following result.

Theorem 4 *Suppose $C^T W^{(k)} = I_p$. Then, the pair $(W^{(k+1)}, Z^{(k+1)})$ satisfies the update (14) if and only if it satisfies*

$$J(w^{(k)})w^{k+1} = ({Z^{(k+1)}}^T \otimes I_n) w^{(k+1)}. \tag{19}$$

and $C^T W^{(k+1)} = I_p$.

Since J is not block diagonal, (19) cannot be easily devectorized as was done for (16). For the special case $p = 1$, we directly identify that (19) reduces to

$$J(w^{(k)})w^{(k+1)} = \lambda^{(k+1)} w^{(k+1)}. \tag{20}$$

Similar to (17), (20) is a standard eigenvalue problem and we can compute a next iterate with a solver for standard eigenvalue problem. It directly suggests that the matrix $A(w^{(k)})$ in the SCF iteration can be viewed as an approximation of the Jacobian matrix, and in order to obtain faster convergence it can be better to use $J(w^{(k)})$, or approximations thereof. This in turn leads to quadratic convergence and in contrast to Newton's method, the method converges in one step for a linear problem. It is superior to Newton's method for problems that are close to being linear, as we prove in Section 5.3.

4.3 Further implementation aspects

Although the theory above provides us with explicit ways ((19) and (17)) to implement our implicitly formulated methods, they do not automatically enforce the constraint $C^T W^{(k+1)} = I_p$. To this end, we compute an intermediate eigenvector eigenpair (Y, Z) and add an additional step in our algorithms to enforce orthogonality and compute $V^{(k+1)}$. Note that C can be chosen freely. Without substantial additional computation, we can impose

$${V^{(k+1)}}^T V^{(k+1)} = I_p, \tag{21}$$

that is, $C = V^{(k+1)}$. We compute a thin QR factorization of Y to obtain $V^{(k+1)}$ and perform a similarity transformation using the R matrix to compute $S^{(k+1)}$. This need not be performed in every step and can be done just once at the end, since only $V^{(k+1)}$ is required for the algorithm to execute the next step. We prefer the normalization condition (21) over a constant C because numerical linear algebra folklore tells us that orthogonal matrices have better numerical stability properties. Combining (19) and (17) with this normalization step leads to Algorithm 1 and Algorithm 2.

Remark 1 (*Selection of eigenvectors*) The iterates of both algorithms will depend upon which of the p eigenvectors are used to construct $V^{(k+1)}$ in each step. The best choice is usually application dependent. For example, in quantum chemistry, one would select the eigenvectors corresponding to the p smallest eigenvalues. This is because only the p smallest eigenvalues are of interest, which correspond to the so-called occupied states [29].

In the the case $p = 1$, specifically when we apply the method to the Gross-Pitaevskii equation, another strategy is natural. Suppose we use the algorithms in the second stage, as part of the two-stage approach (outlined in Section 1), where the first stage is done with a gradient flow method. We can use the eigenvector computed in the first stage as an initial guess for the iteration, and the eigenvalue as a target in the selection procedure. More precisely, if the gradient flow returns x which is an approximation of the ground state, we can compute the corresponding eigenvalue approximation with the Rayleigh quotient $\sigma = x^T A(x)x$. In each iteration of Algorithm 1 or Algorithm 2, we can select the eigenvector corresponding to the eigenvalue closest to σ.

Yet another application with $p = 1$ occurs in hierarchical spectral clustering using the p-Laplacian [7]. However, in contrast to the Gross-Pitaevskii application, one is interested in computing the eigenpair corresponding to the second smallest eigenvalue. By construction, the p-Laplacian always has a zero eigenvalue and hence, the strategy in this case will be to select the smallest non-zero eigenpair.

Algorithm 1 J-version.

```
input  : $V^{(0)} \in \mathbb{R}^{n\times p}$ such that ${V^{(0)}}^T V^{(0)} = I$
output: $(V_*, S_*) \in \mathbb{R}^{n\times p} \times \mathbb{R}^{p\times p}$ that satisfies (1)
for $k = 0, 1, \ldots,$ until convergence do
    Find $(Y, Z)$ such that
        $J(v^{(k)})y = (Z^T \otimes I_n)y.$
    Compute thin QR factorizaton of $Y$ to get $V^{(k+1)}$.
        $Y = V^{(k+1)}R.$
    Compute a similarity transformation of Z: $S^{(k+1)} = R^{-1}ZR.$
end
```

Algorithm 2 A-version.

```
input  : $V^{(0)} \in \mathbb{R}^{n\times p}$ such that ${V^{(0)}}^T V^{(0)} = I$
output: $(V_*, S_*) \in \mathbb{R}^{n\times p} \times \mathbb{R}^{p\times p}$ that satisfies (1)
for $k = 0, 1, \ldots,$ until convergence do
    Find $(Y, Z)$ such that
        $A(V^{(k)})Y = YZ.$
    Compute thin QR factorizaton of $Y$ to get $V^{(k+1)}$.
        $Y = V^{(k+1)}R.$
    Compute a similarity transformation of Z: $S^{(k+1)} = R^{-1}ZR.$
end
```

In this setting, we only provide an exact direct computation formulation for (19) when $p = 1$, which has many important applications, most notably in computing numerical solutions to the Gross-Pitaevskii equation (see Section 6.2). When $p > 1$, we are not aware of any exact direct computation formulation, but we illustrate that if we resort to approximate solution methods, we do obtain similar attractive convergence properties. The approximate solution approach is illustrated with a specific example in the simulations section (Section 6.3).

5 Convergence theory

5.1 Local convergence of Algorithm 2

Since Algorithm 2 is equivalent to SCF as shown in Theorem 3, the convergence can be described in the setting of SCF. There has been extensive study of convergence of SCF and its acceleration in the last fifty years. Several results exist in the literature, as mentioned in Section 1. In general, SCF exhibits linear local convergence when it converges. Convergence can be characterized in terms of gaps [35] (see also [34, Theorem 3.1]). Rather than reviewing the details of the convergence results, we refer the reader to the general characterizations, e.g., in [3, 19, 20, 35] and the references therein.

5.2 Local convergence of Algorithm 1

Due to our inexact Newton viewpoint, the convergence of Algorithm 1 can be characterized using results in the rich literature on inexact Newton methods. Quadratic local convergence can be proved using theorems in [11].

Theorem 5 *Let* $(W^{(k)}, Z^{(k)})$*,* $k = 1, \ldots,$ *be a sequence of pairs satisfying* (14)*. Then,* $C^T W^{(k)} = I_p$ *for* $k = 2, \ldots,$*. If the sequence converges monotonically to a solution* (W_*, Z_*) *to* (10)*, and the Jacobian given by* (12) *in invertible, then it converges with the same convergence order as Newton's method.*

We refer the reader to the Appendix for the proof.

5.3 Single step analysis

As we illustrate in the examples, the implicit methods often work well in general and in particular for close-to-linear problems. This is intuitively natural since both implicit methods converge in one step if we apply it to linear problems.

This can be further characterized, by considering one step of the method applied to a problem parameterized by a parameter α, where $\alpha = 0$ corresponds to a linear problem. For this analysis, we consider the model problem

$$A(v) = A_0 + \alpha K(v).$$

Let $\begin{bmatrix} v_0, \lambda_0 \end{bmatrix}^T$ be an initial guess and $\begin{bmatrix} v_+, \lambda_+ \end{bmatrix}^T$ be the result of (for the moment) any of the two algorithms. We introduce three functions

$$G_\beta\left(\begin{bmatrix} \lambda \\ v \\ \alpha \end{bmatrix}\right) = \begin{bmatrix} (A_0 + \alpha P_\beta)v - \lambda v \\ c^T v - 1 \end{bmatrix}$$

where β can be $\beta = *$, $\beta = A$, and $\beta = J$. The values of P (where we dropped the parameters for notational convenience) denote the nonlinearity

$$P_* = K(v_*) \tag{22a}$$
$$P_A = K(v_0) \tag{22b}$$
$$P_J = \left(\frac{d}{dv}(K(v)v)\right)_{v=v_0}. \tag{22c}$$

These functions correspond to the residual for the exact solution ($\beta = *$), one step of Algorithm 1 ($\beta = J$) and one step of Algorithm 2 ($\beta = A$). Note that $v_+(0) = v_*(0)$ and $\lambda_+(0) = \lambda_*(0)$, since $\alpha = 0$ corresponds to a linear eigenvalue problem, respectively.

We can apply the implicit function theorem for all three functions, and express the first $n+1$ variables in terms of the third variable α in a neighborhood of the solution, if the associated Jacobian is non-singular. The Jacobian given (12) is now assumed to be non-singular in the solution. The exact solution can then be expanded as

$$\begin{bmatrix} v_*(\alpha) \\ \lambda_*(\alpha) \end{bmatrix} = \begin{bmatrix} v_*(0) \\ \lambda_*(0) \end{bmatrix} - \alpha \begin{bmatrix} A_0 - \lambda_*(0)I & -v_*(0) \\ c^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} K(v_*(0))v_*(0) \\ 0 \end{bmatrix} + \mathcal{O}(\alpha^2) \tag{23}$$

whereas both $\beta = A$ and $\beta = J$ can be expanded as

$$\begin{bmatrix} v_+(\alpha) \\ \lambda_+(\alpha) \end{bmatrix} = \begin{bmatrix} v_+(0) \\ \lambda_+(0) \end{bmatrix} - \alpha \begin{bmatrix} A_0 - \lambda_*(0)I & -v_*(0) \\ c^T & 0 \end{bmatrix}^{-1} \begin{bmatrix} c_\beta \\ 0 \end{bmatrix} + \mathcal{O}(\alpha^2) \tag{24}$$

where $c_A = P_a v_*(0)$ and $c_J = P_j v_*(0)$.

The first terms in the Taylor expansion of the next iterate and the exact iterate as a function of the parameterization of the nonlinearity are equal. Therefore,

$$\begin{bmatrix} v_+(\alpha) \\ \lambda_+(\alpha) \end{bmatrix} - \begin{bmatrix} v_*(\alpha) \\ \lambda_*(\alpha) \end{bmatrix} = \mathcal{O}(\alpha)$$

meaning that the accuracy of one step is of the order of magnitude of the nonlinear term. Moreover, the coefficient is proportional to $\|K(v_*(0))v_*(0) - P_\beta v_*(0)\|$.

6 Simulations

6.1 Scalar nonlinearity

The theory and methods are first illustrated with a reproducible example where $p = 1$. We consider (2) with

$$A(v) = A_0 + \alpha \sin\left(\frac{v^T A_2 v}{v^T v}\right) A_1 \tag{25}$$

and

$$A_0 = \frac{1}{10}\begin{pmatrix} 10 & 21 & 13 & 16 \\ 21 & -26 & 24 & 2 \\ 13 & 24 & -26 & 37 \\ 16 & 2 & 37 & -4 \end{pmatrix}, \quad A_1 = \frac{1}{10}\begin{pmatrix} 20 & 28 & 12 & 32 \\ 28 & 4 & 14 & 6 \\ 12 & 14 & 32 & 34 \\ 32 & 6 & 34 & 16 \end{pmatrix},$$
$$A_2 = \frac{1}{10}\begin{pmatrix} -14 & 16 & -4 & 15 \\ 16 & 10 & 15 & -9 \\ -4 & 15 & 16 & 6 \\ 15 & -9 & 6 & -6 \end{pmatrix}$$

and $\alpha \in \mathbb{R}$. Note that A in (25) satisfies Assumption 1 if we select

$$B(X) = A_0 + \alpha \sin\left(\frac{c^T XBXc}{c^T Xc}\right) A_1$$

for essentially any $c \in \mathbb{R}^n$. This specific example appears in [16, Section 3.3] and J is explicitly given by

$$J(v) = A(v) + 2\alpha \frac{\cos\left(\frac{v^T A_2 v}{v^T v}\right)}{(v^T v)^2} A_1 v((v^T v)v^T A_2 - (v^T A_2 v)v^T).$$

We solve four instances of this problem generated by four different values of α, that is $\alpha = 0, 0.5, 1$, and 5, corresponding to four different weights of the nonlinear term. To all of these instances, we apply Algorithm 1 (using (20)), Algorithm 2, the J-Inverse iteration (from [16]), and Newton's method with initial guess $v_0 = \begin{pmatrix} 1, & 1, & 1, & 1 \end{pmatrix}^T$. In Fig. 1, we see the error history of all three methods for all four values of α. The error is computed as $\|v_k - v_*\|$, where v_* is the reference solution.

We observe linear convergence for Algorithm 2 and quadratic convergence for Algorithm 1 as predicted by the theory in Section 5. Both implicit methods are competitive, at least for small values of α. For higher values of α, the number of iterations required to enter the regime of quadratic convergence increases for both Algorithm 1 and Newton's method. This example illustrates a simple case when Algorithm 1 is a better choice than Newton's method, although both methods converge quadratically. We observe linear convergence for J-Inverse iteration as predicted by [16, Theorem 3.1].

In Fig. 2, we visualize the implications of the theory in Section 5.3 by plotting the single step errors for all four methods. It is clear that the single step error is linear in α, as expected from (23) and (24). The predicted line is plotted using the coefficent $\|C(v_*(0))v_*(0) - P_\beta v_*(0)\|$. This illustrates an advantage of the proposed methods for small α.

The implicit algorithms derived in this paper may be used in combination with an additional globalization strategy to make them more robust with respect to the

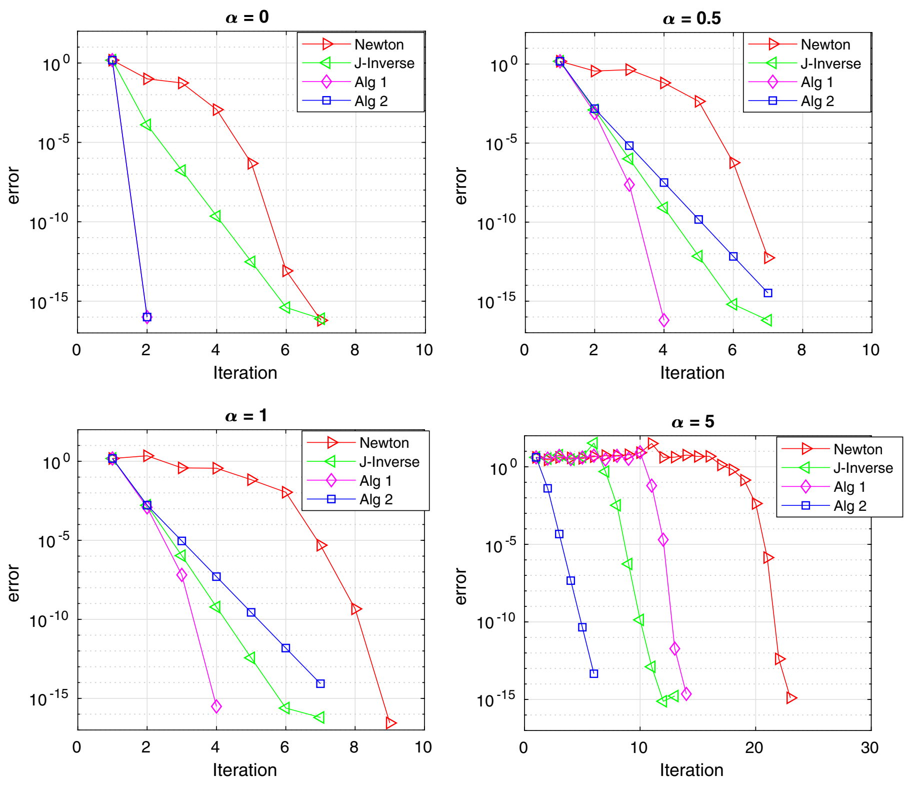

Fig. 1 Algorithm 1, Algorithm 2, J-Inverse iteration, and Newton's method for different α

choice of initial guess. For example, in many applications, it is common to use the Armijo rule [2] to find a suitable step size along the update direction computed by a Newton-based method. Figure 3 is an illustration of how the convergence basin can be enlarged when we use the Armijo rule. More precisely, we apply Algorithm 1 with many different initial guesses which are obtained by perturbing the second and third components (represented by the horizontal and vertical axes of the plots) of the reference eigenvector $v_* \approx \begin{pmatrix} 0.2107 & -0.6730 & -0.3909 & 0.5915 \end{pmatrix}^T$. The figure contains two separate contour plots of the natural logarithm of the residual after ten iterations. The plot on the left is obtained without using any globalization strategy. The one on the right is obtained by combining Algorithm 1 with one sub-step that implements the Armijo rule. The addition of the Armijo rule clearly benefits Algorithm 1 by enlarging the set of initial guesses that lead to convergence.

6.2 Computing the ground state of bosons

The Gross-Pitaevskii equation (GPE) is a nonlinear PDE obtained by a Hartree-Fock approximation (see [28]) of the Schrödinger equation. It describes the ground state

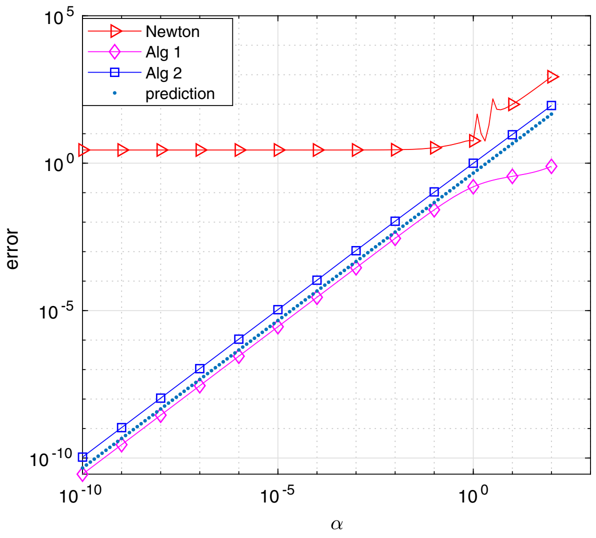

Fig. 2 Dependence of single step error on α

of identical bosons in a quantum system. We consider the case of a rotating Bose-Einstein condensate on the domain $\mathbb{D} = (-L, L) \times (-L, L)$. In this case, the GPE for the wave function $\Psi : \mathbb{R}^2 \to \mathbb{C}$ under an external potential $V : \mathbb{R}^2 \to \mathbb{R}$ is

$$\left(-\frac{1}{2}\Delta - i\Omega\frac{\partial}{\partial\phi}\Psi(x, y) + V(x, y)\right) + b|\Psi(x, y)|^2\Psi(x, y) = \lambda\Psi(x, y), \quad (x, y) \in \mathbb{D}. \tag{26}$$

Here, $\frac{\partial}{\partial\phi} = y\frac{\partial}{\partial x} - x\frac{\partial}{\partial y}$. The scalar b is a constant indicating the strength of interaction between the bosons and Ω is the angular velocity of rotation. We choose the boundary condition $\Psi(x, y) = 0$ for $(x, y) \in \partial D$.

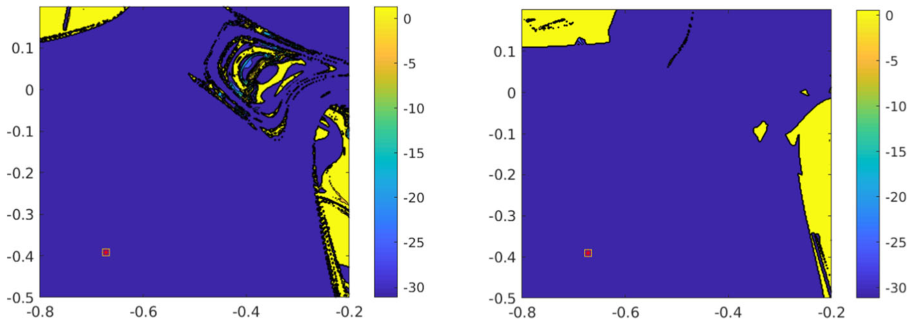

Fig. 3 Convergence basin for Algorithm 1, without (left) and with application of the Armijo rule (exact solution marked with a red “*”)

We perform a central difference discretization of (26) using a uniform grid of $N+2$ points along each dimension with grid spacing Δx. Details are in [16, Section 5.1]. This leads to a problem of size $n = 2N^2$ with

$$\begin{aligned} A(v) &= \begin{pmatrix} \text{Re } \tilde{A}_0 & -\text{Im } \tilde{A}_0 \\ \text{Im } \tilde{A}_0 & \text{Re } \tilde{A}_0 \end{pmatrix} + \frac{\gamma}{v^T v} B(v), \\ B(v) &= \begin{pmatrix} \text{diag}(v_1) + \text{diag}(v_2) & 0 \\ 0 & \text{diag}(v_1) + \text{diag}(v_2) \end{pmatrix}^2, \\ v &= \begin{pmatrix} v_1, & v_2 \end{pmatrix}' \end{aligned}$$

where $\tilde{A}_0$ is the discretization of the linear operator $-\frac{1}{2}\Delta - i\Omega \dfrac{\partial}{\partial \phi} + V(x, y)$ and $\gamma = b(\Delta x)^{-2}$. Note that both $v_1, v_2 \in \mathbb{R}^{N^2}$ and $v_1 + iv_2$ give the vectorization of Ψ evaluated at the interior points. We have

$$\begin{aligned} J(v) = &\begin{pmatrix} \text{Re } \tilde{A}_0 & -\text{Im } \tilde{A}_0 \\ \text{Im } \tilde{A}_0 & \text{Re } \tilde{A}_0 \end{pmatrix} \\ &+ \tfrac{\gamma}{v^T v}\left[\begin{pmatrix} 3\text{diag}(v_1)^2 + \text{diag}(v_2)^2 & 2\text{diag}(v_1)\text{diag}(v_2) \\ 2\text{diag}(v_1)\text{diag}(v_2) & \text{diag}(v_1)^2 + 3\text{diag}(v_2)^2 \end{pmatrix} - \tfrac{2}{v^T v} B(v) v v^T \right]. \end{aligned} \tag{27}$$

We now apply and compare the performance of Algorithm 1 and J-Inverse iteration with J as defined by (27).

As seen from Fig. 4, Algorithm 1 converges in much fewer iterations as compared to the J-Inverse iteration. Since each iteration of Algorithm 1 involves the solution of a linear eigenvalue problem, we see that it performs worse when we measure the cumulative computation times for each iteration. This is because the J-Inverse iteration solves a linear system in each step (Fig. 5).

Since one step of both Algorithm 1 and Algorithm 2 requires the solution of a standard eigenvalue problem, we need to select an appropriate eigenpair at each step. Special attention is needed in the selection in this problem. We select a new iterate in a way that minimizes the difference between two iterates. More precisely, we choose $\delta \in (0, 1)$ and select all eigenpairs which correspond to eigenvalue within a radius δ of a given target. We then do a least squares fitting to find the linear combination of these eigenvectors which is closest to the previous iterate. This is needed due to the fact that the problem has highly clustered eigenvalues.

6.3 Invariant subspace

We consider

$$A(V) = A_0 + \alpha \text{ diag } ({A_0}^{-1}\text{diag}(h(VV^T))) \tag{28}$$

where A_0 is the discrete 1D Laplacian. This problem is a simplified version of problems that frequently occur in electronic structure calculations when we discretize the Hartree-Fock approximation of the Schrödinger equation. See [35] for a discussion of the problem type and convergence results of SCF applied to (28).

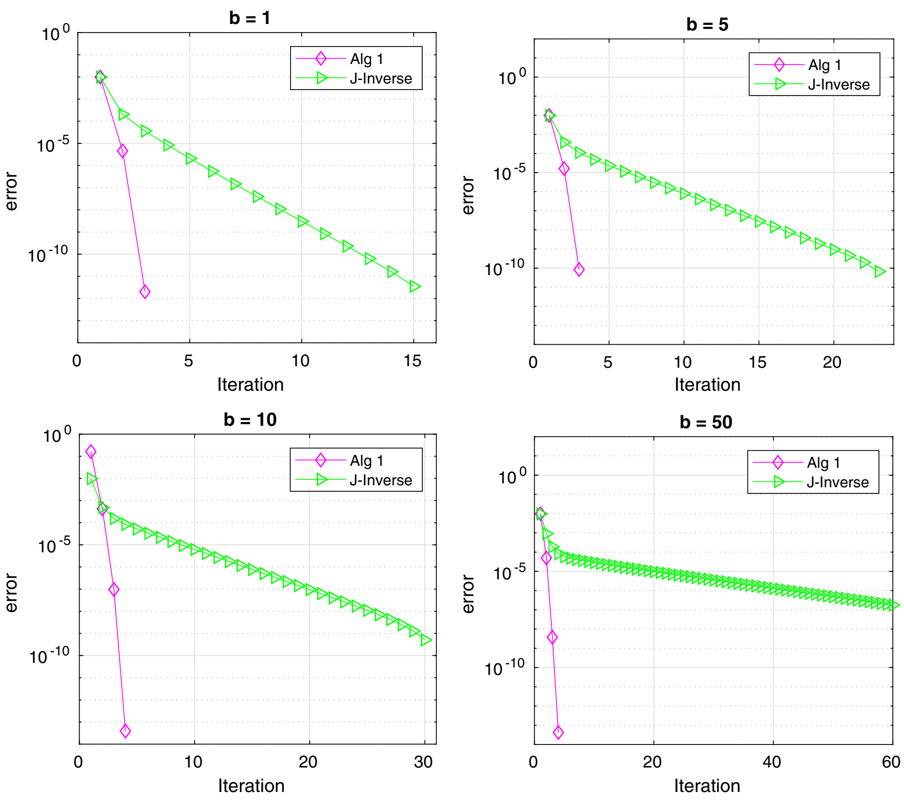

Fig. 4 Algorithm 1 and J-Inverse iteration applied to GPE for different b

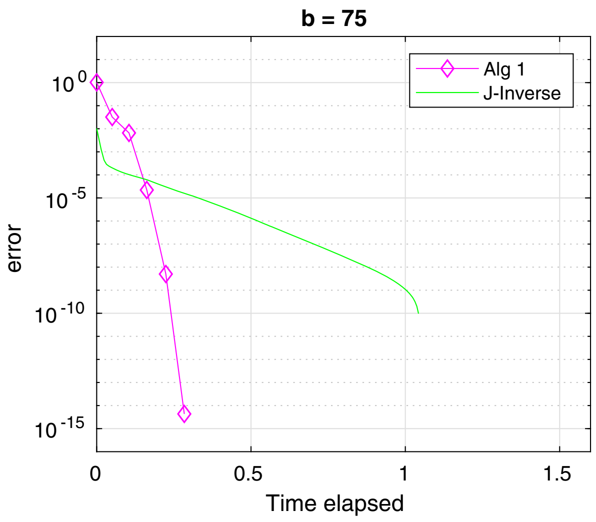

Fig. 5 A comparison of computation times for Algorithm 1 and J-Inverse iteration

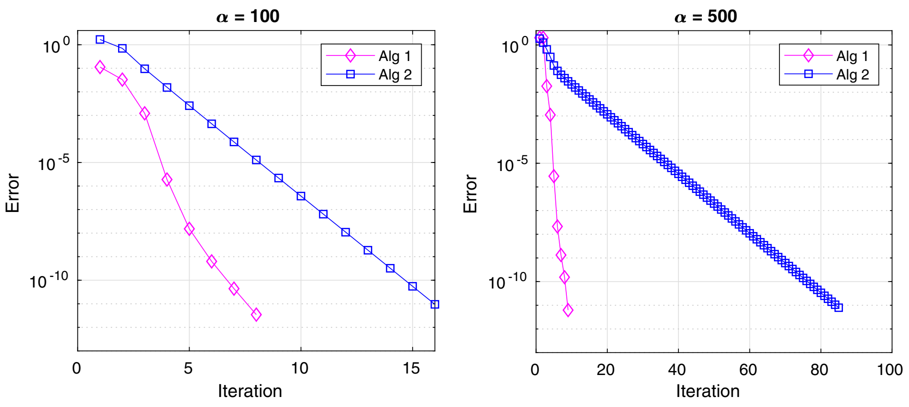

Fig. 6 Algorithm 1 and Algorithm 2 for different α

If we let e_i denote the the ith column of the identity matrix I_n and $E_{i,j} = e_i e_j^T$, then

$$A(V) = A_0 + \alpha \sum_{i=1}^{n} (e_i^T h(VV^T)e_i)\mathrm{diag}(A_0^{-1} E_{i,i}).$$

The derivation and some computational aspects of $J(V)$ are contained in the (publicly available) extended technical report [17, Appendix B], which is omitted for brevity.

The implementation of Algorithm 2 follows directly from (17). We also illustrate the importance of the implicit formulation of Algorithm 1 by inexactly solving (19). We do this with the optimization subroutine *fminsearch*. It provides us a way to test Algorithm 1 for relatively small examples as a proof of concept. We use $n = 10$ and $p = 3$ and apply Algorithm 1 and Algorithm 2 for two different values of α.

In Fig. 6, we observe that Algorithm 2 converges linearly and Algorithm 1 has much faster convergence, with an initial quadratic phase. The number of iterations required for convergence increases with increase in α, as expected from the single step analysis of Section 5.3. The initial quadratic phase is succeeded by an asymptotic slowdown, which can be attributed to the inexact solution of the update (19).

7 Conclusions and outlook

This paper shows that taking an inexact and implicit Newton approach towards deriving algorithms for problems with eigenvector nonlinearities leads to new algorithmic insights. Using this approach, we derive two algorithms. Algorithm 2 is shown to be the widely used SCF algorithm. This result shows a connection between Newton's method and the SCF algorithm which was previously unknown. Algorithm 1 is a new algorithm, to the best of our knowledge.

We prove that Algorithm 1 exhibits quadratic local convergence. Both Algorithm 1 and Algorithm 2 have favorable convergence properties for problems that are close to being linear, as shown by the single step analysis of Section 5.3. Numerical simulations for the Gross-Pitaevskii equation in Section 6.2 show that Algorithm 1 is a competitive algorithm for the $p = 1$ case. The $p > 1$ example in Section 6.3 shows that Algorithm 1 converges faster than Algorithm 2 even when we solve the update (19) inexactly.

There are several improvements of the SCF algorithm. Some of these techniques may be interpretable from an implicit viewpoint as well. For instance, acceleration schemes such as DIIS [25] might be seen as an inexact Newton algorithm. This could be combined with other convergence theories to gain further understanding of DIIS. Another direction that can be explored is to develop application-specific strategies to solve the (19) for $p > 1$ or approximate solution techniques that lead to superlinear convergence.

Appendix. Proof of Theorem 5

Proof Let F'_k be the Jacobian of F evaluated in the iterate k. In the notation of [11], we introduce a residual denoted r_k, corresponding to the difference between a Newton step and an inexact Newton step:

$$F'_k \begin{bmatrix} \Delta w^{(k)} \\ \Delta z^{(k)} \end{bmatrix} = -F^{(k)} + r_k. \tag{29}$$

Then, subtracting (14) from (29) the residual becomes

$$r_k = \begin{bmatrix} 0 & -I_p \otimes (W^{(k)} - W^{(k+1)}) \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta v^{(k)} \\ \Delta s^{(k)} \end{bmatrix}$$

Using the Cauchy-Schwarz inequality, we have

$$\|r_k\| \leq \left\| \begin{bmatrix} 0 & -I_p \otimes (W^{(k)} - W^{(k+1)}) \\ 0 & 0 \end{bmatrix} \right\| \left\| \begin{bmatrix} \Delta w^{(k)} \\ \Delta z^{(k)} \end{bmatrix} \right\| \leq \mathcal{O}\left(\left\| \begin{bmatrix} \Delta w^{(k)} \\ \Delta z^{(k)} \end{bmatrix} \right\|^2 \right) \tag{30}$$

By the assumption of monotonic convergence, we have,

$$\left\| \begin{bmatrix} \Delta w^{(k)} \\ \Delta z^{(k)} \end{bmatrix} \right\| \leq 2 \left\| \begin{bmatrix} w^{(k)} - w_* \\ z^{(k)} - z_* \end{bmatrix} \right\| = \mathcal{O}\left(\left\| \begin{bmatrix} w^{(k)} - w_* \\ z^{(k)} - z_* \end{bmatrix} \right\| \right). \tag{31}$$

From the implicit function theorem (e.g., the formulation in [26, Theorem 9.28]) and the assumption about the invertibility of the Jacobian at the solution, we get that

$$\begin{bmatrix} w^{(k)} \\ s^{(k)} \end{bmatrix} = \begin{bmatrix} w_* \\ s_* \end{bmatrix} + (F'_*)^{-1} F^{(k)} + \mathcal{O}(\|F^{(k)}\|^2). \tag{32}$$

where F'_* is the Jacobian of F evaluated in the solution. The combination of (30), (31), and (32) leads to

$$\|r_k\| = \mathcal{O}(\|F^{(k)}\|^2).$$

By [11, Theorem 3.3], the proof is complete. □

Acknowledgements We thank Prof. Daniel Kressner for comments on a preliminary version of this manuscript.

Funding Open access funding provided by Royal Institute of Technology.

References

1. Altmann, R., Henning, P., Peterseim, D.: The J-method for the Gross-Pitaevskii eigenvalue problem. Tech. rep., Univ. Augsburg. arXiv:2009.09022 (2019)
2. Armijo, L.: Minimization of functions having Lipschitz continuous first partial derivatives. Pacific J. Math. **16**(1), 1–3 (1966). https://doi.org/10.2140/pjm.1966.16.1
3. Bai, Z., Li, R.C., Lu, D.: Optimal convergence rate of self-consistent field iteration for solving eigenvector-dependent nonlinear eigenvalue problems. Tech. rep. arXiv:1908.00333 (2020)
4. Bai, Z., Lu, D., Vandereycken, B.: Robust Rayleigh quotient minimization and nonlinear eigenvalue problems. SIAM J. Sci. Comput. **40**(5), A3495–A3522 (2018). https://doi.org/10.1137/18M1167681
5. Bao, W., Du, Q.: Computing the ground state solution of Bose-Einstein condensates by a normalized gradient flow. SIAM J. Sci. Comput. **25**(5), 1674–1697 (2004). https://doi.org/10.1137/S1064827503422956
6. Borgatti, S.P., Everett, M.G.: Models of core/periphery structures. Soc. Networks **21**(4), 375–395 (2000). https://doi.org/10.1016/S0378-8733(99)00019-2
7. Bühler, T., Hein, M.: Spectral clustering based on the graph p-Laplacian. In: Proceedings of the 26th International Conference on Machine Learning, pp. 81–88 (2009)
8. Cancès, E., Bris, C.L.: On the convergence of SCF algorithms for the Hartree-Fock equations. M2AN, Math. Model. Numer. Anal. **34**(4), 749–774 (2000). https://doi.org/10.1051/m2an:2000102
9. Cancès, E., Kemlin, G., Levitt, A.: Convergence analysis of direct minimization and self-consistent iterations. Tech. rep. arXiv:2004.09088 (2020)
10. Claes, R., Jarlebring, E., Meerbergen, K., Upadhyaya, P.: Linearizability of eigenvector nonlinearities. Tech. rep. arXiv:2105.10361 (2021)
11. Dembo, R.S., Eisenstat, S., Steihaug, T.: Inexact Newton methods. SIAM J. Numer. Anal. **19**, 400–408 (1982). https://doi.org/10.1137/0719025
12. Edelman, A., Arias, T.A., Smith, S.T.: The geometry of algorithms with orthogonality constraints. SIAM J. Matrix Anal. Appl. **20**(2), 303–353 (1998)
13. Hein, M., Bühler, T.: An inverse power method for nonlinear eigenproblems with applications in 1-spectral clustering and sparse PCA. In: Advances in Neural Information Processing Systems 23, pp. 847–855 (2010)
14. Higham, N.: Functions of Matrices. SIAM . https://doi.org/10.1137/1.9780898717778 (2008)
15. Jarlebring, E., Koskela, A., Mele, G.: Disguised and new quasi-Newton methods for nonlinear eigenvalue problems. Numer. Algorithms **79**, 331–335 (2018). https://doi.org/10.1007/s11075-017-0438-2
16. Jarlebring, E., Kvaal, S., Michiels, W.: An inverse iteration method for eigenvalue problems with eigenvector nonlinearities. SIAM J. Sci. Comput. **36**(4), A1978–A2001 (2014). https://doi.org/10.1137/130910014
17. Jarlebring, E., Upadhyaya, P.: Implicit algorithms for eigenvector nonlinearities. Tech. rep., KTH Royal Institute of Technology. arXiv:2002.12805 (2002)
18. Levitt, A.: Convergence of gradient-based algorithms for the Hartree-Fock equations. ESAIM Math. Model. Numer. Anal. **46**, 1321–1336 (2012). https://doi.org/10.1051/m2an/2012008

19. Liu, X., Wang, X., Wen, Z., Yuan, Y.: On the convergence of the self-consistent field iteration in Kohn-Sham density functional theory. SIAM J. Matrix Anal. Appl. **35**(2), 546–558 (2014). https://doi.org/10.1137/130911032
20. Liu, X., Wen, Z., Wang, X., Ulbrich, M., Yuan, Y.: On the analysis of the discretized Kohn-Sham density functional theory. SIAM J. Numer. Anal. **53**(4), 1758–1785 (2015). https://doi.org/10.1137/140957962
21. Mehrmann, V., Voss, H.: Nonlinear eigenvalue problems: a challenge for modern eigenvalue methods. GAMM Mitteilungen **27**, 121–152 (2004)
22. Mercado, P., Tudisco, F., Hein, M.: Spectral clustering of signed graphs via matrix power means. In: Proceedings of the 36th International Conference on Machine Learning, pp. 4526–4536 (2019)
23. Peters, G., Wilkinson, J.: Inverse iterations, ill-conditioned equations and Newton's method. SIAM Rev. **21**, 339–360 (1979). https://doi.org/10.1137/1021052
24. Pulay, P.: Convergence acceleration of iterative sequences - the case of SCF iteration. Chem. Phys. Lett **73**(2), 393–398 (1980). https://doi.org/10.1016/0009-2614(80)80396-4
25. Rohwedder, T., Schneider, R.: An analysis for the DIIS acceleration method used in quantum chemistry calculations. J. Math. Chem. **49**(9), 1889–1914 (2011). https://doi.org/10.1007/s10910-011-9863-y
26. Rudin, W.: Principles of Mathematical Analysis. 3rd ed McGraw-Hill (1976)
27. Ruhe, A.: Algorithms for the nonlinear eigenvalue problem. SIAM J. Numer. Anal. **10**, 674–689 (1973). https://doi.org/10.1137/0710059
28. Saad, Y., Chelikowsky, J.T., Shontz, S.M.: Numerical methods for electronic structure calculations of materials. SIAM Rev. **52**(1), 3–54 (2010). https://doi.org/10.1137/060651653
29. Szabo, A., Ostlund, N.S.: Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory. Dover Publications (1996)
30. Tapia, R.A., Dennis, J.E., Schäfermeyer, J.P.: Inverse, shifted inverse, and Rayleigh quotient iteration as Newton's method. SIAM Rev. **60**(1), 3–55 (2018). https://doi.org/10.1137/15M1049956
31. Tudisco, F., Higham, D.J.: A nonlinear spectral method for core–periphery detection in networks. SIAM J. Math. Data. Sci. **1**(2), 269–292 (2019). https://doi.org/10.1137/18M1183558
32. Unger, G.: Convergence orders of iterative methods for nonlinear eigenvalue problems. Springer, Berlin. https://doi.org/10.1007/978-3-642-30316-6_10 (2013)
33. Unger, H.: Nichtlineare Behandlung von Eigenwertaufgaben. Z. Angew. Math. Mech. **30**, 281–282 (1950). https://doi.org/10.1002/zamm.19500300839. English translation: http://www.math.tu-dresden.de/~schwetli/Unger.html
34. Upadhyaya, P., Jarlebring, E., Rubensson, E.H.: A density matrix approach to the convergence of the self-consistent field iteration. Numer. Alg. Control Optimization, Accepted for publication (2020)
35. Yang, C., Gao, W., Meza, J.C.: On the convergence of the self-consistent field iteration for a class of nonlinear eigenvalue problems. SIAM J. Matrix Anal. Appl. **30**(4), 1773–1788 (2009). https://doi.org/10.1137/080716293
36. Zhao, Z., Bai, Z.J., Jin, X.Q.: A Riemannian Newton algorithm for nonlinear eigenvalue problems. SIAM J. Matrix Anal. Appl. **36**(2), 752–774 (2015)