Induced Dimension Reduction Method to Solve the Quadratic Eigenvalue Problem

Astudillo, R.; van Gijzen, M.B.

DOI
10.1007/978-3-319-57099-0_20

Publication date
2017

Document Version
Final published version

Published in
Numerical Analysis and Its Applications - 6th International Conference, NAA 2016

Citation (APA)
Astudillo, R., & van Gijzen, M. B. (2017). Induced Dimension Reduction Method to Solve the Quadratic Eigenvalue Problem. In I. Dimov, I. Faragó, & L. Vulkov (Eds.), *Numerical Analysis and Its Applications - 6th International Conference, NAA 2016: Revised Selected Papers* (pp. 203-211). (Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Vol. 10187 LNCS). Springer. https://doi.org/10.1007/978-3-319-57099-0_20

Induced Dimension Reduction Method to Solve the Quadratic Eigenvalue Problem

R. Astudillo(✉) and M.B. van Gijzen

Delft Institute of Applied Mathematics, Delft University of Technology,
Mekelweg 4, 2628 CD Delft, The Netherlands
{R.A.Astudillo,M.B.vanGijzen}@tudelft.nl

Abstract. In this work we are interested in the numerical solution of the Quadratic Eigenvalue Problem (QEP)

$$(\lambda^2 M + \lambda D + K)\mathbf{x} = \mathbf{0},$$

where M, D, and K are given matrices of order N. Particularly, we study the applicability of the IDR(s) for eigenvalues to solve QEP. We present an IDR(s) algorithm that exploits the special block structure of the linealized QEP to compute its eigenpairs. To this end we incorporate ideas from Second Order Arnoldi method proposed in [3].

Keywords: Quadratic Eigenvalue Problem · Induced Dimension Reduction

1 Introduction

In this work we are interested in solving the Quadratic Eigenvalue Problem (QEP), i.e., find a subset of pairs (λ, $\mathbf{x}$), where $\lambda \in \mathbb{C}$ and $\mathbf{x} \in \mathbb{C}^N$ such that,

$$(\lambda^2 M + \lambda D + K)\mathbf{x} = \mathbf{0}, \tag{1}$$

where M, D, and K are (sparse) matrices of order N often referred as mass, damping, and stiffness matrices, respectively. The QEP appears in different areas like vibration analysis, dynamical systems, or stability of flows in fluid mechanics (see [8] and their references within).

One of the most common options to solve the quadratic eigenvalue problem (1) is to linearized it to an standard eigenvalue problem. First, problem (1) can be written as a generalized eigenvalue problem, i.e.

$$C\mathbf{y} = \lambda G\mathbf{y}, \tag{2}$$

where

$$C = \begin{bmatrix} -D & -K \\ I & 0 \end{bmatrix}, \quad \text{and} \quad G = \begin{bmatrix} M & 0 \\ 0 & I \end{bmatrix}.$$

I. Dimov et al. (Eds.): NAA 2016, LNCS 10187, pp. 203–211, 2017.
DOI: 10.1007/978-3-319-57099-0_20

Second, if the matrix M is not singular, (2) can be rewritten as standard eigenvalue problem,

$$A\mathbf{y} = \lambda\mathbf{y}, \tag{3}$$

with

$$A = \begin{bmatrix} -M^{-1}D & -M^{-1}K \\ I & 0 \end{bmatrix}. \tag{4}$$

It is easy to check that the eigenvalues of A and the eigenvalues of (1) are related by:

$$\mathbf{y}_i = \begin{bmatrix} \lambda_i \mathbf{x}_i \\ \mathbf{x}_i \end{bmatrix}. \tag{5}$$

Then, one can apply any eigensolver software for the standard eigenvalue (3) and obtain approximate solutions of the quadratic eigenvalue problem (1). This approach has two main disadvantages. First, it solves a standard eigenvalue problem of double the dimension of the original quadratic eigenvalue problem. Second, some properties of the matrices M, D, and K are lost during the linealization; for example, matrices M, D and K can be symmetric positive definite (SPD) matrices but the matrix A does not keep the SPD property.

To overcome the disadvantages of using the linealization (3), the authors in [3] propose a method called Second Order Arnoldi (SOAR), which is a modification of the Arnoldi method [1]. By exploiting the block structure of the matrix A, the SOAR method uses approximately half of the memory of the classical Arnoldi method applied to the problem (3). Also and more importantly, this method preserves essential structures and properties of the matrices involved.

The Arnoldi method has as main drawback its demanding computational requirements. In this contribution, we study the Induced Dimension Reduction Method for eigenvalue problem [2] to solve the Quadratic Eigenvalue Problem as an alternative to the Arnoldi method.

This document is organized as follow. Section 2 presents an introduction to the Induced Dimension Reduction method for solving linear system of equations. In Sect. 3, we present how the IDR(s) method has been adapted to solve the standard eigenvalue problem, and in Sect. 4, we present an IDR(s) to solve the Quadratic Eigenvalue problem, using ideas from SOAR. In Sect. 5, we conduct numerical experiments to illustrate the numerical behavior of the IDR(s) for QEP. Section 6 presents the conclusions and remarks of this work.

2 Induced Dimension Reduction Method - IDR(s)

IDR(s) was presented originally in [7], as a short recurrences iterative Krylov method to solve large and sparse systems of linear equations,

$$A\mathbf{x} = \mathbf{b}. \tag{6}$$

The IDR(s) method is based on the following theorem,

Theorem 1 (IDR(s) theorem). *Let A be any matrix in $\mathbb{C}^{N\times N}$, and let $P = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_s]$ be an $N \times s$ matrix with s linear independent columns. Let $\{\mu_j\}$ be a sequence in $\mathbb{C}$. With $\mathcal{G}_0 \equiv \mathbb{C}^N$, define*

$$\mathcal{G}_{j+1} \equiv (A - \mu_{j+1} I)(\mathcal{G}_j \cap P^{\perp}) \qquad j = 1, 2 \ldots, \tag{7}$$

where $P^{\perp}$ represents the orthogonal complement of P. If $P^{\perp}$ does not contain an eigenvector of A, then, for all $j = 0, 1, 2 \ldots,$ the following hold

1. $\mathcal{G}_{j+1} \subset \mathcal{G}_j$, and
2. dimension($\mathcal{G}_{j+1}$) $<$ dimension($\mathcal{G}_j$) unless $\mathcal{G}_j = \{\mathbf{0}\}$.

Proof. See [7].

The subspaces $\mathcal{G}_j$ for $j = 0, 1, 2, \ldots$ are shrinking and nested subspaces. IDR(s) uses recurrences of size s to create an approximated solution $\mathbf{x}_{k+1}$ forcing its corresponding residual vector $\mathbf{r}_{k+1} = \mathbf{b} - A\mathbf{x}_{k+1}$ to be in the subspace $\mathcal{G}_{j+1}$. Using the fact that $\mathcal{G}_{j^*} = \{\mathbf{0}\}$ for some j^*, the residual will become zero and IDR(s) will obtain the solution of (6).

3 IDR(s) to Solve the Eigenvalue Problem

Several methods to compute a subset of eigenpairs $(\lambda_i, \mathbf{x}_i)$ of a large and sparse matrix $A \in \mathbb{C}^{N\times N}$ rely on the construction of a standard Hessenberg relation of the form,

$$AU_m = U_m H_m + \mathbf{f}\,\mathbf{e}_m^T, \tag{8}$$

where H_m is an upper Hessenberg matrix of order m (much smaller than N), $U_m = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m] \in \mathbb{C}^{N\times m}$ is a basis for the Krylov subspace $\mathcal{K}_m(A, \mathbf{u}_1)$ with $\mathbf{u}_1 \neq \mathbf{0}$, $\mathbf{f} \in \mathbb{C}^N$, and $\mathbf{e}_m$ is the m-th canonical vector. It can be proved that the eigenpairs of A can be approximated by $(\hat{\lambda}_j, U_m\hat{\mathbf{y}}_j)$, where $(\hat{\lambda}_j, \hat{\mathbf{y}}_j)$ are the eigenpairs of the smaller matrix H_m.

Two examples of well-known methods to construct a Hessenberg relation and approximate eigenpairs, are the Lanczos [5] and Arnoldi [1] method. While the Lanczos method is suitable when the coefficient matrix is symmetric, in the case of unsymmetric matrices this method might suffer from numerical instability. For this reason, the Arnoldi method is the most common option to build a Hessenberg relation for unsymmetric matrices. The Arnoldi method explicitly build an orthogonal basis for the Krylov subspace and because of this the work and storage per iteration grow with the number of iterations.

In [4] and later in [2] the IDR(s) method was adapted to build a Hessenberg relation and approximate eigenpairs of unsymmetric large matrices. The Hessenberg relations based on the IDR(s) method keep the computational work (almost) constant per iteration. Next, we review of how to obtain an IDR-Hessenberg relation.

IDR creates a vector in $\mathcal{G}_{j+1}$ with the assumption that $s+1$ vectors are already in $\mathcal{G}_j$ namely $\{\mathbf{w}_{k-i}\}_{i=0}^{s}$. A new vector $\mathbf{w}_{k+1}$ in the subspace $\mathcal{G}_j$ can be written as,

$$\mathbf{w}_{k+1} = (A - \mu_{j+1} I)\left(\mathbf{w}_k - \sum_{i=1}^{s} \beta_i \mathbf{w}_{k-i}\right), \tag{9}$$

where the coefficient β_i are computed via the solution of the following $s \times s$ system of linear equation,

$$P^T[\mathbf{w}_{k-1}, \mathbf{w}_{k-2}, \dots, \mathbf{w}_{k-s}]\mathbf{c} = P^T \mathbf{w}_k \quad \text{where} \quad \mathbf{c} = [\beta_1, \dots, \beta_s]^T.$$

It is possible to rewrite (9) as,

$$A\left(\sum_{i=0}^{s} \beta_i \mathbf{w}_{k-i}\right) = \mathbf{w}_{k+1} - \mu_{j+1} \sum_{i=0}^{s} \beta_i \mathbf{w}_{k-i}, \tag{10}$$

with $\beta_0 = -1$. From the equation above, the authors in [4] constructed a generalized Hessenberg relation,

$$A W_m \hat{U}_m = W_m \hat{H}_m + \mathbf{w}\mathbf{e}_m^T, \tag{11}$$

where $\hat{U}_m$ is an upper triangular matrix and $\hat{H}_m$ is an upper banded Hessenberg matrix. The banded matrix pencil $(\hat{H}_m, \hat{U}_m)$ is called the Sonneveld pencil. The eigenvalues of this pencil are divided into two sets: $\{\mu_k\}_{i=1}^{t}$ where t is the number of subspaces $\mathcal{G}_j$ created, and the approximations to the eigenvalues of A or Ritz values $\{\theta_k\}_{i=t}^{m}$. In [2], the authors construct a standard Hessenberg relation using IDR(s),

$$A W_m = W_m H_m + \mathbf{w}\mathbf{e}_m^T = W_{m+1} \bar{H}_m, \tag{12}$$

where the $i-$th column of the upper Hessenberg matrix H_m is defined as,

$$\mathbf{h}_i = \left(\begin{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \\ -\mu_{j+1} \begin{bmatrix} c_1 \\ \vdots \\ c_s \end{bmatrix} \\ \mu_{j+1} \\ 1 \end{bmatrix} + \sum_{\ell=1}^{s} c_\ell \mathbf{h}_{i-\ell} \right). \tag{13}$$

and this matrix has the same eigenvalues as the pencil $(\hat{H}_m, \hat{U}_m)$ obtained from (11) in [4]. At this point, one can apply directly IDR(s) for eigenvalues to the problem (3). Especially in applications where only the eigenvalues are needed, IDR(s) can be used as a short-recurrences method to create H_m (see Eq. (9) and (13)) and approximate the eigenvalues of (1).

4 IDR(s) to Solve the Quadratic Eigenvalue Problem

Bai and Su in [3] proposed a special version of the Arnoldi algorithm for the QEP. This so called second order Arnoldi algorithm (SOAR) exploits the structure of (3)–(5) to reduce the memory requirements for the Arnoldi method by a factor of two. In this section we examine how the ideas underlying in SOAR can be incorporated in IDR to obtain a second order IDR (SOIDR) algorithm.

4.1 Second Order IDR(s)

One can exploit the block structure of the $2N \times 2N$ matrix A in Eq. (4) for the creation a standard Hessenberg relation (12). Let us consider Eq. (12), with the matrix W_m rewritten in two block matrices of size $N \times m$ as,

$$W_m = \begin{bmatrix} W_m^{(U)} \\ W_m^{(L)} \end{bmatrix}, \tag{14}$$

Then Eq. (12) can be written as,

$$-M^{-1}DW_m^{(U)} - M^{-1}KW_m^{(L)} = W_m^{(U)}H_m + \mathbf{w}_{m+1}^{(U)}\mathbf{e}_m^T \tag{15}$$

$$W_m^{(U)} = W_m^{(L)}H_m + \mathbf{w}_{m+1}^{(L)}\mathbf{e}_m^T. \tag{16}$$

From Eq. (16), and assuming that the first column vector ($\mathbf{w}_1$) of the matrix W_m has the following pattern

$$\mathbf{w}_1 = \begin{bmatrix} \mathbf{u} \\ \mathbf{0} \end{bmatrix}, \qquad \text{with } \mathbf{u} \neq \mathbf{0} \in \mathbb{C}^N,$$

we have (using the Matlab subindex notation),

$$W_m^{(U)} = W_{m+1}^{(L)}\bar{H}_m = W_{m+1}^{(L)}(:, 2:m+1)\bar{H}_m(2:m+1, 1:m). \tag{17}$$

The Eq. (15) can be rewritten as,

$$-M^{-1}DW_m^{(U)} - M^{-1}KW_m^{(U)}T_m = W_m^{(U)}H_m + \mathbf{w}_{m+1}^{(U)}\mathbf{e}_m^T, \tag{18}$$

where

$$T_m = \begin{bmatrix} \mathbf{0} & \bar{H}_m(2:m, 1:m-1)^{-1} \\ 0 & \mathbf{0} \end{bmatrix}. \tag{19}$$

Equations (18) and (19) suggest a formula to compute the column vectors of the matrix $W_m^{(L)}$ as a linear combination of the column vector of $W_m^{(U)}$. Algorithm 1 shows a possible implementation of these ideas. This method needs only half of the memory of the classical Arnoldi method applied to the matrix A. The memory requirements of SOIDR are equivalent to those from SOAR [3].

Algorithm 1. SOIDR(s) for solving the QEP.

```
1: Given $s \in \mathbb{N}$, $P \in \mathbb{R}^{N\times s}$, $M$, $D$, and $K$.
2: Run SOAR to obtain $W \in \mathbb{C}^{n\times s+1}$ and $H \in \mathbb{C}^{s+1\times s}$, s.t.
   $$A \begin{bmatrix} W^{(U)} \\ W^{(L)} \end{bmatrix}_s = \begin{bmatrix} W^{(U)} \\ W^{(L)} \end{bmatrix}_{s+1} \bar{H}_s.$$
3: $T_s = \bar{H}_s(2:s, 1:s-1)^{-1}$
4: for $i = s+1, \ldots, m$ do
5:     if $i$ is multiple of $s+1$ then
6:         Choose the parameter $\mu_j$ for the subspace $\mathcal{G}_j$ .
7:     end if
8:     Solve $(P^T[\mathbf{w}_{i-s}^{(U)}, \mathbf{w}_{i-s+1}^{(U)}, \ldots, \mathbf{w}_{i-1}^{(U)}])\mathbf{c} = P^T\mathbf{w}_i^{(U)}$.
9:     $\mathbf{v} = \mathbf{w}_i^{(U)} - \sum_{\ell=1}^{s} \beta_\ell \mathbf{w}_{i-\ell}^{(U)}$.
10:    Compute the latest column of $W_i^{(L)}$ as $\mathbf{v}^{(L)}$ using (17) and (19).
11:    $\mathbf{w}_{i+1}^{(U)} = -M^{-1}(D\mathbf{v} + K\mathbf{v}^{(L)}) - \mu_j\mathbf{v}$.
12:    Create the $i$-th column of $H$ according to (13).
13:    Update $T_i$ using (19).
14:    $W_{i+1}^{(U)} = [\mathbf{w}_1^{(U)}, \mathbf{w}_2^{(U)}, \ldots, \mathbf{w}_i^{(U)}, \mathbf{w}_{i+1}^{(U)}]$.
15: end for
16: Compute the eigenpairs $\{(\lambda_i, \mathbf{z}_i)\}_{i=1}^m$ s.t. $H_m\mathbf{z}_i = \lambda_i\mathbf{z}_i$.
17: return $\{(\lambda_i, W_m^{(U)}\mathbf{z}_i)\}_{i=1}^m$.
```

It is important to mention that the SOIDR(s) algorithm and IDR(s) for eigenvalues generate the same Ritz values for the same input parameters. The low-memory IDR(s) for eigenvalues algorithm is a short-recurrences method that uses $2s+2$ vectors of dimension $2N$ and it is only possible obtain approximations to the eigenvalues. While SOIDR(s) can obtain approximation of the eigenpairs, this is a more expensive algorithm. SOIDR(s) uses long-recurrences in the step (10) of Algorithm 1 and the complete upper part of W_m needs to be stored.

5 Numerical Experiments

Experiment 1: The purpose of this example is to compare the convergence of Arnoldi, SOAR, and SOIDR for the exterior eigenvalues of the QEP. The matrices M, D, and K are random sparse of order 400. Figure 1 shows a comparison between the errors of the Ritz values generated by Arnoldi, SOAR, and SOIDR(4) or 35 Ritz values after 40 matrix vector products (Fig. 2 shows the Ritz values computed).

Experiment 2: In our second experiment we measure the execution times for SOAR, SOIDR(4), and IDR(4) for eigenvalues [2]. We only compute a set of 10 the eigenvalues of the (1) and the matrices M, D and K are random matrices of size 6000×6000. Table 1 shows the CPU times for each method.

Experiment 3: This example was presented in [6], and models the propagation of sound waves in a room with five solid walls and one wall of a sound-absorbing material,

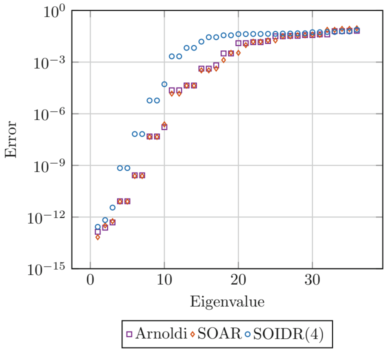
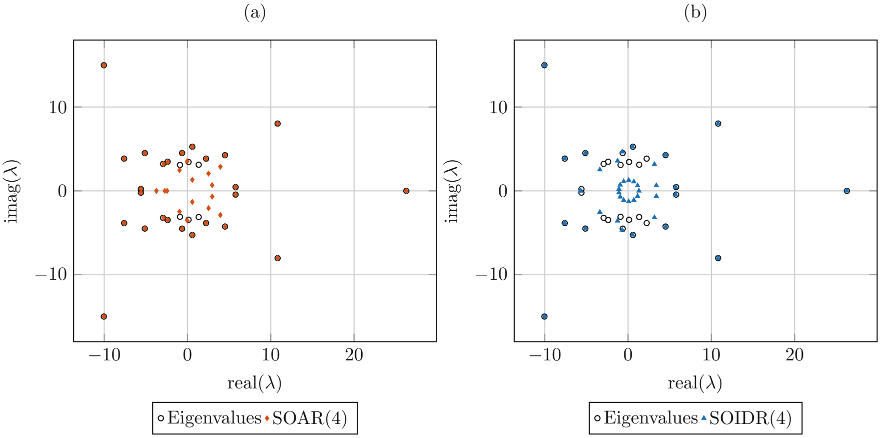


Fig. 1. *Experiment 1:* Convergence for 35 Ritz values after 40 matrix vector products. One can see a similar convergence behavior, however some Ritz values of the SOIDR(4) have a larger error.


(a)

(b)

Fig. 2. *Experiment 1:* (a) Exterior eigenvalues and their approximation by SOAR. (b) Exterior eigenvalues and their approximation by SOIDR(4).

Table 1. *Experiment 2:* Execution time comparison for SOAR, SOIDR(4), and IDR(4) after 40 matrix vector products.

Method	Time [s]
SOAR	3.67
SOIDR(4)	3.78
IDR(4)	2.41

$$\frac{\lambda}{c^2}p - \triangle p = 0 \quad \text{in } [-2.0,\, 2.0]^3 \tag{20}$$

where c is the speed of sound (340 meter/second) and the boundary conditions are,

$$\frac{\partial p}{\partial n} = 0 \qquad \text{for the solid walls,} \tag{21}$$

and,

$$\frac{\partial p}{\partial n} = -\frac{\lambda}{cZ_n}p \qquad \text{for the absorbing wall.} \tag{22}$$

Selecting an impedance $Z_n = 0.2 - 1.5i0$, this problem has an analytical eigenvalue $-5.19 + 217.5i$. We discretized Eqs. (20)–(22) using finite element and obtain matrices of order 1681. Figure 3 shows the evolution of the error of the Ritz values generated by SOAR and SOIDR(2).

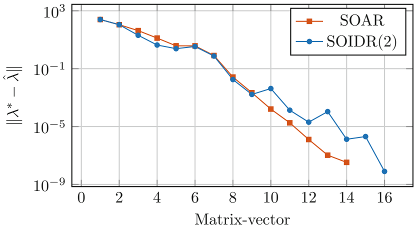


Fig. 3. *Experiment 3:* Error convergence for SOAR and SOIDR(2) to the known eigenvalues $\lambda^* = -5.19 + 217.5i$.

6 Conclusions

We have developed a second order IDR algorithm (SOIDR) based on the ideas underlying SOAR. In contrast to IDR for the standard eigenvalue problem, SOIDR is not short recurrence. The memory requirements for SOIDR are comparable to that of SOAR. Compared to SOIDR, SOAR is for our test cases the preferred method, since it exhibits a faster convergence. However, more numerical test are needed.

References

1. Arnoldi, W.E.: The principle of minimized iterations in the solution of the matrix eigenvalue problem. Q. Appl. Math. **9**, 17–29 (1951)
2. Astudillo, R., van Gijzen, M.B.: A restarted induced dimension reduction method to approximate eigenpairs of large unsymmetric matrices. J. Comput. Appl. Math. **296**, 24–35 (2016)
3. Bai, Z., Su, Y.: SOAR: a second-order arnoldi method for the solution of the quadratic eigenvalue problem. SIAM J. Matrix Anal. Appl. **26**(3), 640–659 (2005)

4. Gutknecht, M.H., Zemke, J.-P.M.: Eigenvalue computations based on IDR. SIAM J. Matrix Anal. Appl. **34**(2), 283–311 (2013)
5. Lanczos, C.: An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. J. Res. Natl. Bur. Stand. **45**(4), 255–281 (1950)
6. Sleijpen, G.L.G., van der Vorst, H.A., van Gijzen, M.B.: Quadratic eigenproblems are no problem. SIAM News **29**(7), 8–19 (1996)
7. Sonneveld, P., van Gijzen, M.B.: IDR(s): a family of simple and fast algorithms for solving large nonsymmetric systems of linear equations. SIAM J. Sci. Comput. **31**(2), 1035–1062 (2008)
8. Tisseur, F., Meerbergen, K.: The quadratic eigenvalue problem. SIAM Rev. **43**(2), 235–286 (2001)