Jacobi-Davidson method for Polynomial Eigenvalue Problems

Master Thesis

by

Ammar Al Jammaz

Heinrich-Heine-Universität, Germany

Department of Mathematics
applied Mathematics Group

October 2008
Acknowledgement

I am very grateful to my supervisor Prof. Dr. Marlis Hochbruck for her guidance and many helpful suggestions during the preparation of this thesis. I thank Dominik Löchel for his comments and suggestions on reading a draft of this work. Finally, my wife whose constant encouragement was integral to the completion of this work. Her love, affection and humour enabled me to switch off and relax when I needed to. This thesis is dedicated to her.
Contents

1 Jacobi-Davidson Method 7
   1.1 Some basics .............................................. 7
   1.2 Davidson method ......................................... 10
   1.3 Jacobi’s orthogonal component correction method (JOCC)  12
   1.4 JD method ................................................. 14
       1.4.1 Jacobi-Davidson method as Rayleigh quotient iteration 16
       1.4.2 Solution of the correction equation .................... 17
       1.4.3 Examples ............................................. 18

2 The Quadratic Eigenvalue Problem (QEP) 23
   2.1 The Generalized Eigenvalue Problem ....................... 23
       2.1.1 The QZ Algorithm .................................... 24
   2.2 QEP ............................................................ 26
       2.2.1 The solution of QEP .................................. 35

3 Higher Order Polynomial Eigenvalue Problems (PEP) 39
   3.1 JD method for PEP .......................................... 40
   3.2 Solving the correction equation ........................... 43
   3.3 Numerical example ......................................... 47

A 53
   A.1 A list of basic iterative methods .......................... 53
   A.2 iterative solvers for linear systems ....................... 56
       A.2.1 CG for Hermitian positive definite systems ........... 56
       A.2.2 GMRES ................................................. 58
List of Algorithms

1. Rayleigh quotient iteration ........................................ 8
2. Rayleigh-Ritz approximation ........................................ 8
3. Subspace iteration ...................................................... 9
4. Davidson .............................................................. 12
5. JOCC ................................................................. 13
7. J-D method for polynomial eigenvalue problem ..................... 42
8. ILU ................................................................. 46
9. biCG($z, b, B, \epsilon, k_{max}$) Algorithm ................................. 47
10. QR Algorithm .......................................................... 54
11. QR Algorithm with shift ................................................. 54
12. Arnoldi Algorithm ....................................................... 55
13. Lanczos Algorithm ....................................................... 56
14. CG Algorithm .......................................................... 57
15. GMRES Algorithm ....................................................... 59
Chapter 1

Jacobi-Davidson Method

In this chapter we will discuss the Jacobi-Davidson method for the iterative computation of a few of the extremal eigenvalues of a symmetric matrix and their associated eigenvectors. It was suggested in 1996 by Sleijpen and Van der Vorst [14]. We want to consider the Jacobi-Davidson method which is also an subspace algorithm with increasing dimension of the subspace. It combines ideas from algorithms by Jacobi and Davidson. Again, the Rayleigh-Ritz procedure is applied to a sequence of subspaces of increasing dimension. But, in the Jacobi-Davidson method the constructed subspaces are no longer Krylov subspaces. Instead, in each step, the subspace is expanded with an orthogonal correction to a Ritz vector in order to obtain a better approximation of an eigenvector.

1.1 Some basics

We have the following standard eigenvalue problem:

\[ Ax = \lambda x \quad \text{where} \quad A \in \mathbb{C}^{n \times n}, 0 \neq x \in \mathbb{C}^n, \lambda \in \mathbb{C}. \]

**Definition 1.1.1.** We call that the matrix \( A \in \mathbb{C}^{n \times n} \) sparse if the number of non-zero elements is small compared to the number of zero entries.

We want to solve the last problem by iterative methods, which are methods based on matrix-vector multiplications using the original sparse matrices so that the sparse matrices storage and structure can be used advantage. Hence, subspace algorithms are suitable for large sparse matrices. Different subspace methods are distinguished from the way the subspaces are generated. We can work with subspaces of both fixed and variable dimension.
1. Single vector iterations (Rayleigh quotient iteration):

this method uses a variable shift and this shift is chosen to be the Rayleigh quotient \( \theta = x^H Ax \).

Algorithm 1 Rayleigh quotient iteration

1: Input \( A, x^{(0)}(\|x^{(0)}\|= 1), i_{\text{max}} \).
2: Compute \( \theta^{(0)} = x^{(0)H} Ax^{(0)} \).
3: for \( i = 1 \) to \( i_{\text{max}} \) do
4: \( y^{(i)} = (A - \theta^{(i-1)}I)^{-1}x^{(i-1)} \).
5: \( x^{(i)} = \frac{y^{(i)}}{\|y^{(i)}\|} \).
6: \( \theta^{(i)} = x^{(i)H} Ax^{(i)} \).
7: end for
8: Output \( \theta^{(i_{\text{max}})}, x^{(i_{\text{max}})} \).

2. Subspace iteration with fixed dimension (Rayleigh-Ritz procedure):

We use a so called Galerkin approach to solve it:

- Suppose we have a subspace \( K \) which generates an orthonormal basis \( v_1, \cdots, v_k \).
- With \( V_k = [v_1, \cdots, v_k] \in \mathbb{C}^{n \times k} \), a matrix with orthonormal columns.

Galerkin condition: find \( u \) and \( \theta \) such that:

\[
\begin{align*}
r := Au - \theta u \perp \{v_1, \cdots, v_k\} = K, u \in K.
\end{align*}
\]

or

\[
\begin{align*}
V_k^H AV_k s = \theta s, \quad \text{where} \quad u = V_k s.
\end{align*}
\]

Algorithm 2 Rayleigh-Ritz approximation

1: Compute an orthonormal basis \( \{v_i\}_{i=1,\cdots,k} \) of the subspace \( K \).
   Let \( V = [v_1, \cdots, v_k] \).
2: Compute \( H_k = V_k^H AV_k \).
3: Compute the eigenvalues of \( H_k \) and select the \( m \) desired ones \( \theta_i, i = 1, \cdots, m \), where \( m \leq k \).
4: Compute the eigenvectors \( s_i, i = 1, \cdots, m \) of \( H_k \) associated with \( \theta_i, i = 1, \cdots, m \) and the corresponding approximate eigenvectors of \( A \), \( u_i = V s_i, i = 1, \cdots, m \).
The orthogonal projection of $A$ onto $K$ is given by $P_K A P_K$, where $P_K = V_k V_k^H$.

**Definition 1.1.2.** $(\theta, V_k s) = (\theta, u)$ is called Ritz pair associated with the subspace (search space) $K = \text{span}\{v_1, \cdots, v_k\}$. $(\theta, u)$ with Ritz residual approximates the eigenpair $(\lambda, x)$ of $A$.

3. Subspace iteration with increasing dimension are (Arnoldi, Lanczos, Davidson and Jacobi-Davidson)

In practice we want $k \ll n$. All subspace algorithms with increasing dimension have the following structure in common:

1. Generate an orthonormal system $v_1, \cdots, v_k$ and we wish to add $v_{k+1}$.
2. Find an expansion vector $t$ for the subspace $V_k$.
3. Expand the subspace by orthogonalisation of $t$ against $v_1, \cdots, v_k$ (via modified Gram-Schmidt).
4. Solve the slightly bigger projected problem :

$$ V_{k+1}^H A V_{k+1} s = \theta s. $$

**Algorithm 3** Subspace iteration

1. Chose initial subspace $V_1$.
2. for $j = 1, 2$ do
3. $W_j \rightarrow AV_j$, $H_j \rightarrow V_j^H W_j$
4. Compute desired eigenpair $(\theta, s)$ of $H_j$, with $\| s \| = 1$
5. $u \rightarrow V_j s$
6. $r \rightarrow Au - \theta u$
7. Stop if satisfied
8. Compute an expansion vector $t$
9. Expand subspace $V_{j+1} \rightarrow MGS[V_j, t]$
10. end for

There are different choices for the expansion vector $t$, assuming we have the initial subspace $\text{span}\{v_1\}$.

- **a.** Arnoldi’s method: $t = Av_k$, then $H := V_k^H A V_k$ is upper Hessenberg.
- **b.** Lanczos’s method: $H := V_k^H A V_k$ is tridiagonal (only for $A = A^H$).
CHAPTER 1. JACOBI-DAVIDSON METHOD

- For both methods the search space \( \text{span}\{V_k\} = \text{span}\{v_1, Av_1, \cdots, A^{k-1}v_1\} \) is a so called Krylov subspace.

- Arnoldi and Lanczos favour extremal eigenvalues.

- Shift-and-Invert Arnoldi: \( t = (A - \alpha I)^{-1}v_k \) approach favours eigenvalues close to \( \alpha \).

Theorem 1.1.1. (Bauer-Fike) Let \((\tilde{\lambda}, \tilde{u})\) be an approximate eigenpair with \( \| \tilde{u} \| = 1 \), and \( r = A\tilde{u} - \tilde{\lambda}\tilde{u} \) (residual vector). Assume \( A \) is diagonalizable \( A = XDX^{-1} \), with \( D \) diagonal. Then

\[
\exists \lambda \in \Lambda(A) : | \lambda - \tilde{\lambda} | \leq \text{cond}_2(X) \| r \|_2.
\]  

Remark 1.1.1.

- Note that for hermitian matrices \( X = Q \) becomes unitary, and hence (1.1) becomes

\[
\exists \lambda \in \Lambda(A) : | \lambda - \tilde{\lambda} | \leq \| r \|_2,
\]

where \( \text{cond}_2(X) = 1 \).

- If \( E \) is a perturbation to \( A \) then for any eigenvalue \( \tilde{\lambda} \) of \( A + E \) there is an eigenvalue \( \lambda \) of \( A \) such that:

\[
| \lambda - \tilde{\lambda} | \leq \text{cond}_2(X) \| E \|_2.
\]

Theorem 1.1.2. Let \( A \in \mathbb{C}^{n\times n} \) and a unit vector \( u \) be given. For any scalar \( \alpha \) define the residual vector \( r := Au - \alpha u \) and let \( \mathcal{K} = \{ E : (A - E)u = \alpha u \} \) then

\[
\min \| E \|_2 = \| r \|, \text{ for any } E \in \mathcal{K}.
\]

1.2 Davidson method

The Davidson method is an iterative subspace method for calculating eigenvalues. It is particularly well-suited for diagonally dominant matrices \([3]\). Let \( \text{span}\{v_1, \cdots, v_k\} \) be a search subspace, in which the matrix \( A \) has a Ritz value \( \theta_k \) and a corresponding Ritz vector \( u_k \). The matrix \( V_k \in \mathbb{C}^{n\times k} \) stores the orthogonal basis vectors \( v_1, \cdots, v_k \).

The idea behind the Davidson method is the following:
The algorithm proceeds in the identical manner as Lanczos algorithm, building by the \( j \)-th step an orthogonal subspace spanned by the columns of \( V_k \) onto which \( A \) is projected, yielding the projected matrix \( H_k = V_k^H A V_k \). The dominant eigenpair \((\theta, s)\) of \( H_k \) is computed, and the Ritz vector \( u = V_k s \) is obtained, where \( \|s\| = 1 \). Compute the residual of the Ritz approximation

\[
    r = Au - \theta u
\]

and, if it is small enough for acceptance of the Ritz pair \((\theta, u)\) then stop.

**Remark 1.2.1.** Note that \( \|u\| = \|s\| = 1 \) since \( V_k \) is orthogonal. Also

\[
    u^H r = s^H V_k^H (AV_k s - \theta V_k s) = s^H (H s - \theta s) = 0. \tag{1.2}
\]

Equation (1.2) and \( \|u\| = 1 \) imply : \( \theta = u^H A u \).

If \( \|r\| \leq \varepsilon \) (\( \varepsilon \) is the tolerance for the stopping criteria) then there must be an eigenvalue \( \lambda \) of \( A \) such that \( |\lambda - \theta| \leq \varepsilon \). Indeed the Wielandt theorem [12] states that if \( A \) symmetric (hermitian) and \( E \) is a perturbation to \( A \) (symmetric or not) then

\[
    |\lambda(A) - \lambda(A + E)| \leq \|E\|.
\]

Pick \( \lambda(A) = \lambda \) and \( E = -ru^H \). Then \((A + E)u = Au - r \|u\|^2 = \theta u \), i.e. \( \theta \) is an eigenvalue of \( A + E \) and

\[
    |\lambda - \theta| = \|E\| = \|r\|\|u\| \leq \varepsilon.
\]

In this case the pair \((\theta, u)\) is accepted as the maximal eigenpair of \( A \) and the iteration is stopped. However, if convergence has not been realized, it is necessary to expand our basis, and it is at this point that the Davidson portion of the algorithm, i.e., preconditioning, takes place. It is not the vector \( u \) which is orthonormalized against the columns of the \( V_k \) in order to get the next basis vector, instead solve for \( t \) from:

\[
    (D_A - \theta I)t = r
\]

where \( D_A \) is the diagonal matrix consisting of the diagonal elements of \( A \). The vector \( t \) is orthogonalized against the columns of the \( V_k \), normalized to have norm unity, and the resulting vector is the one which is added to the basis as \( v_{k+1} \):

\[
    V_{k+1} = [V_k | v_{k+1}]
\]
CHAPTER 1. JACOBI-DAVIDSON METHOD

Algorithm 4 Davidson
1: Start: choose an initial vector $v_1$ ($\|v_1\| = 1$).
2: Iterate: Until convergence do:
3: for $j = 1$ to $k$ do
4: Compute $w := Av_j$.
5: Compute $V_j^H w$, the last column of $H_j := V_j^H AV_j$.
6: Compute Ritz vector $u := V_j s$ and associated residual vector $r := Au - \theta u$.
7: Test for convergence. If satisfied Return.
8: Compute $t := - (D_A - \theta I)^{-1} r$.
9: Orthogonalisation of $t$ against $V_j$ via modified Gram-Schmidt $V_{j+1} := MGS([V_j, t])$ (skip when $j = k$).
10: end for
11: Restart: Set $v_1 = u$, and go to Inner loop.

1.3 Jacobi’s orthogonal component correction method (JOCC)

In 1846 Jacobi proposed a method for the calculation of eigenvalues and eigenvectors of symmetric diagonal dominant matrices [10].

- Jacobi considered an eigenvalue problem as a system of linear equation for which his iterative linear solver (the Jacobi or Gauss-Jacobi iteration) might be applicable. Suppose we have a diagonally dominant matrix $A$, of which $a_{11} = \alpha$ is the largest diagonal element. Then $\alpha$ is an approximation for the largest eigenvalue $\lambda$, and $e_1$ is an approximation for the corresponding eigenvector $u$.

- Hence the problem

\[
A \begin{pmatrix} 1 \\ z \end{pmatrix} = \begin{pmatrix} \alpha & b^T \\ b & F \end{pmatrix} \begin{pmatrix} 1 \\ z \end{pmatrix} = \lambda \begin{pmatrix} 1 \\ z \end{pmatrix}
\]

- We are interested in the eigenvalue $\lambda$ that is close in some sense to $\alpha$, and in the corresponding eigenvector $u = (1, z^T)^T$, with component $z$ orthogonal to $e_1$.

- The last problem is equivalent to

\[
\lambda = \alpha + b^T z. \quad (1.3)
\]

\[
(F - \lambda I)z = -b. \quad (1.4)
\]
Jacobi proposed to solve \((F - \lambda I)z = -b\) iteratively by his Jacobi iteration.

- Apply Jacobi iteration with \(z_1 = 0\)

\[
\theta_k = \alpha + b^T z_k
\]

\[(D - \theta_k I)z_{k+1} = (D - F)z_k - b\]

where \(D\) is the diagonal of \(F\).

**Algorithm 5 JOCC**

```plaintext
Function [λ, u] = JOCC(A)
★ extract parts of matrix ★
n = size(A, 1);
F = A(2:n, 2:n); d = diag(F);
b = A(2:n, 1);
α = A(1, 1);
★ initialisation ★
z = zeros(n - 1, 1);
★ iteration ★
while not converged
    λ = α + b^T z; update λ
    z = (d * z - Fz - b) / (d - λ); Jacobi iteration step
end while
u = [1; z];
end function
```

**Remark 1.3.1.** The Jacobi’s approach is looking, at all stages, for the orthogonal complement to the initial approximation \(u_1 = e_1\), not taking into account that better approximation \(u_k = [1, z_k^T]^T\) become available at each state (it may be more efficient to calculate \((u - (u^T u_k)u_k)\).

**Comparison of the Davidson method with JOCC**

1. **Search subspace**: The Davidson method chooses the Ritz pair \((\theta, u)\) as the current eigenpair approximation. \(u\) is in the space spanned by all corrections computed so far. The JOCC method on the other hand calculates the current eigenvector approximation as a simple linear combination of the initial vector \(u_1\), the old eigenvector approximation \(u_k\) and a correction \([0, z]^T\).
CHAPTER 1. JACOBI-DAVIDSON METHOD

2. **Orthogonalisation**: JOCC calculates corrections, that are orthogonal to $u_1 = e_1$. The orthogonalisation results implicitly from the method. In contrast, the corrections calculated by the Davidson method are explicitly orthogonalised to the whole search space $K$ (therefore they are also orthogonal to $v_1$).

3. **Eigenvalue approximations**: The Davidson method uses a Ritz value as the eigenvalue approximation, whereas JOCC computes the eigenvalue approximation according to equation (1.3) using an approximation for $z$.

1.4 JD method

We try to find an optimal expansion of the subspace

- Jacobi and Davidson attempt to find corrections to some initially given eigenvector approximation, they both use fixed operators.
- Let $u$ be an approximation to the eigenvector $x$ of $A$ and $\theta$ the corresponding Ritz value, i.e., $u \in K$, where $K = \text{span}\{v_1, \ldots, v_k\}$ is a low $k$-dimensional subspace (the search space).
- Jacobi-Davidson finds the orthogonal complement for current approximation $u$ w.r.t. the desired eigenvector $x$ of $A$.
- We are interested in seeing what happens in the subspace $\text{span}\{u\}^\perp$.
- Orthogonal projection of $A$ onto the subspace is given by

$$B = PAP = (I - uu^H)A(I - uu^H), \text{ with } \|u\| = 1.$$  

$B$ is the restriction of $A$ to the subspace orthogonal to $u$.

*Note* that for $u = e_1$ we have that $F$ is the restriction of $B$ w.r.t. $e_1^\perp$.

- With $\theta = u^HAu$ it follows that (see Remark 1.2.1):

$$A = B + Auu^H + uu^HA - \theta uu^H.$$  

$\theta = u^HAu$ is the Rayleigh quotient of $u$ and $r = Au - \theta u \in \text{span}\{u\}^\perp$.

- Look for an eigenvalue $\lambda$ of $A$ close to $\theta$, we want to have the correction $t \perp u$ (orthogonal correction) such that:
A(u + t) = \lambda(u + t) or with Bu = 0.

We have:

\[ Bt + \theta u + u(u^H At) + Au - \theta u = \lambda(u + t) \]
\[ (B - \lambda I)t = -r + (\lambda - u^H At - \theta)u, \quad r = Au - \theta u \]

- Since LHS and \( r \) have no component in \( u \) (orthogonal to \( u \)), it follows that the factor for \( u \) must vanish and hence the correction \( t \) satisfies:

\[ (B - \lambda I)t = -r. \]

- We replace \( \lambda \) by the current approximation \( \theta \) just as in Jacobi’s method and Davidson’s method. Then:

\[ (B - \theta I)t = -r, \quad t \perp u. \]

Since \( Pt = t \) and \( Pr = r \) then the JD correction equation is:

\[ P(B - \theta I)Pt = (I - uu^H)(B - \theta I)(I - uu^H)t = -r, \quad t \perp u. \]

i.e., \((A - \theta I)\) is restricted to the orthogonal complement of \( u \).

- Expand the subspace by \( t \) (using MGS) and compute the new Ritz pair in expand subspace.

**Remark 1.4.1.**

1. We are free to use any method for the (approximate) solution of the correction equation and that is not necessary to require diagonal dominance of \( B \) or \( A \).

2. If we approximate \( t \) simply by \( r \), then we formally obtain the same results as with the Arnoldi’s method.

3. If we approximate \( t \) by \((DA - \theta I)^{-1}r\) then we obtain the original Davidson’s method.

4. The correction equation is solved only approximately and its approximate solution is taken for the expansion of the subspace. This is a fundamental difference with the Krylov subspace methods; instead of selecting a subspace as powers of an operator acting on a given starting vector, we select some subspace without Krylov structure and we project the given matrix onto this subspace.
Algorithm 6 J-D method for standard eigenvalue problem

1: Start: choose an initial nontrivial vector $v$.
2: Compute $v_1 = \frac{v}{\|v\|_2}$, $w_1 = Av_1$, $h_{11} = v_1^H w_1$. Set $V_1 = [v_1]$, $W_1 = [w_1]$, $H_1 = [h_{11}]$, $u = v_1$, $\theta = h_{11}$, compute $r = w_1 - \theta u$.
3: Iterate: until convergence do:
   4: for $k = 1$ to $m - 1$ do
      5: Solve (approximately) $(I - uu^H)(A - \theta I)(I - uu^H)t = -r$, $t \perp u$.
      6: Orthogonalisation of $t$ against $V_k$ via modified Gram-Schmidt and expand $V_k$ with this vector to $V_{k+1}$.
      7: Compute $w_{k+1} := Av_{k+1}$ and expand $W_k$ with this vector to $W_{k+1}$.
      8: Compute $V_{k+1}^H w_{k+1}$, the last column of $H_{k+1} := V_{k+1}^H AV_{k+1}$, and $v_{k+1}^H W_{k+1}$, the last row of $H_{k+1}$ (only if $A \neq A^H$).
      9: Compute the largest eigenpair $(\theta, s)$ of $H_{k+1}$ ($\|s\|_2 = 1$).
     10: Compute the Ritz vector $u := v_{k+1}s$, compute $\hat{u} := Au (= W_{k+1}s)$ the associated residual vector $r := \hat{u} - \theta u$.
     11: Test for convergence. Stop if satisfied.
   4: end for
13: Restart: Set $V_1 = [u]$, $W_1 = [\hat{u}]$, $H_1 = [\theta]$, and go to 3.

1.4.1 Jacobi-Davidson method as Rayleigh quotient iteration

We can solve exactly the correction equation

$$(I - uu^H)(A - \theta I)(I - uu^H)t = -r, t \perp u.$$ 

We have $t \perp u$ then $(I - uu^H)t = t$ (we are interested in determining $t \perp u$) and then it follows from correction equation that

$$(A - \theta I)t - \alpha u = -r,$$ 

where $\alpha \in \mathbb{C}$ such that $t \perp u$. Then

$$t = - (A - \theta I)^{-1}r + (A - \theta I)^{-1} \alpha u$$

$$= -u + \alpha (A - \theta I)^{-1}u.$$ 

The solution $t$ is used to expand the search space. When we have a suitable preconditioner $M \simeq (A - \theta I)$ available, then we can compute an approximation $\tilde{t}$ for $t$:

$$\tilde{t} = \alpha M^{-1}u - M^{-1}r.$$ 

The value $\alpha$ is determined by the requirement that $\tilde{t}$ should be orthogonal to $u$: 
1.4. JD METHOD

\[ \alpha = \frac{u^HM_{-1}r}{u^HM_{-1}u}. \]

The last equation leads to several interesting observations:

- If we choose \( \alpha = 0 \), then we obtain the Davidson method (with preconditioner \( M \)). In this case \( \tilde{t} \) will not be orthogonal to \( u \).
- If \( M = (A - \theta I) \), then it reduces to
  \[ t = \alpha(A - \theta I)^{-1}u - u, \quad t \perp u. \]

\( u \) is already in the search space the expansion vector is effectively \((A - \theta I)^{-1}u\), which is the same as for inverse iteration for fixed \( \theta \) and the same as for RQI for \( \theta = u^HAu \).

1.4.2 Solution of the correction equation

Iterative solvers are used to solve the large linear system:

- MINRES (Hermitian), GMRES. If a preconditioned iterative method used to solve the correction equation, then, in each step of the linear solver, a preconditioning equation has to be solved.

If \( M \) is some approximation of \((A - \theta I)\), i.e., \( M^{-1}(A - \theta I) \approx I \). Then the projected matrix on the subspace \( \text{span}\{u\}^\perp \):

\[ \tilde{M} := (I - uu^H)M(I - uu^H). \]

We assume that we use a Krylov solver with initial vector \( t_0 = 0 \) and with left preconditioning for the approximate solution of the correction equation (all iteration vectors for the Krylov solver will be in the subspace orthogonal to \( u \)).

\( \tilde{M} \) can be taken as an approximation of \( \tilde{A} := (I - uu^H)(A - \theta I)(I - uu^H) \) and, in each iterative step, we have to solve an equation of the form \( \tilde{M}z = \tilde{A}v \), where \( v \) is a vector supplied by the Krylov solver.

We will do this in two steps:

- First \( \tilde{A}v = (I - uu^H)(A - \theta I)(I - uu^H)v = (I - uu^H)y \) with \( y = (A - \theta I)v \) since \( u^Hv = 0 \).

- Then solve \( \tilde{M}z = (I - uu^H)y \) and since \( z \perp u \) we have \( \tilde{M}z = y - \beta u \implies z = M^{-1}y - \beta M^{-1}u \) and with \( z \perp u \) we get \( \beta = \frac{u^HM_{-1}y}{u^HM_{-1}u} \) so in each step of the Krylov solver the system \( \tilde{M}y = y \) has to be solved.
CHAPTER 1. JACOBI-DAVIDSON METHOD

Remark 1.4.2.

- The matrix $H_{k+1}$ can be updated at each iteration step from the previous $H_k$ using the relation

$$H_{k+1} = \begin{pmatrix} V_k^H & v_{k+1} \\ v_{k+1}^H \\ \end{pmatrix} A \begin{pmatrix} V_k & v_{k+1} \\ \end{pmatrix}$$

$$= \begin{pmatrix} V_k^H A v_k & V_k^H A v_{k+1} \\ (V_k^H A v_{k+1})^H & v_{k+1}^H A v_{k+1} \\ \end{pmatrix}$$

$$= \begin{pmatrix} (H_k)_{old} & V_k^H A v_{k+1} \\ (V_k^H A v_{k+1})^H & v_{k+1}^H A v_{k+1} \\ \end{pmatrix}$$

If only the upper triangular part of $H$ is stored (by symmetry) only the last column of $H$ needs to be computed at the $(k+1)$-th iteration, i.e.,

$$w = A v_{k+1} \text{ and } H(:, k+1) = \begin{pmatrix} V_k^H A w \\ v_{k+1}^H w \\ \end{pmatrix} = V_k^H w.$$

- As usual, roundoff errors may prevent the orthogonalisation process (orthogonalisation $t$ against $V_k$ via modified Gram-Schmidt) to deliver a computed vector $t$, leading to non-orthogonal $H$ and loss of convergence of the global iteration. To reduce this effect it is common to redo an orthogonalisation sweep of the computed $t$ whenever the angle between $t$ and the subspace generated by the columns of $V_{k+1}$ is small.

Assume that the vector $t$ computed at the end of Step 6 of algorithm 8 is not perfectly orthogonal to $\text{span}\{v_1, \ldots, v_{k+1}\} = \text{range}(V_{k+1})$. Let $\tilde{t} = (I - V_{k+1} V_{k+1}^H) t \perp \text{range}(V_{k+1})$. The angle $0 < \varphi < \frac{\pi}{2}$ between $t$ and $\text{range}(V_{k+1})$ satisfies $\sin \varphi = \frac{\|\tilde{t}\|_2}{\|t\|_2}$. Reorthogonalisation (i.e., computation of $\tilde{t}$, which is used instead of $t$ in the next iteration) is applied if, say, $\sin \varphi \leq \frac{1}{4}$ (i.e., $\varphi \leq 14$ degrees).

1.4.3 Examples

Example 1.4.1. (Let the matrix $A$ of the dimension 1000 have diagonal elements $a_{j,j} = j$. The elements on the sub- and super-diagonal ($a_{j-1,j}$ and $a_{j,j+1}$) are all equal to 0.5, as well as the elements $a_{1000,1}$ and $a_{1,1000}$). For this matrix we compute the largest eigenvalue of $A$ ($\approx 1000.225641$) with the standard Lanczos method, Davidson’s method with diagonal preconditioning ($(D_A - \theta I))^{-1}$, and the Jacobi-Davidson method with the same diagonal
preconditioning. In the table 1.1 we see the effects when we take a starting vector \( u = (0.01, 0.01, \cdots , 0.01, 1)^T \) and this starting vector is still has a large component in the dominating eigenvector. This is reflected by the fact that the Ritz value in the first step of all three methods is equal to 954.695\cdots. The Lanczos process converges slowly. We see the very poor convergence behavior of Davidson’s method (we may interpret the new starting vector as a good starting vector for the perturbed matrix \( A \), this implies that the diagonal preconditioning may not be expected to be a very good preconditioner. The difference with the Jacobi-Davidson method is now quite notable and for this method we observe rather fast convergence again.

| Iteration | Lanczos | Davidson | J-D | | \( \lambda - \theta \) |
|-----------|---------|----------|-----|---|
| 0         | 0.45e+02 | 0.45e+02 | 0.45e+02 | |
| 1         | 0.56e+01 | 0.40e+02 | 0.25e+02 | 0.50e+02 |
| 2         | 0.16e+01 | 0.40e+02 | 0.74e+01 | 0.12e+03 |
| 3         | 0.71e+00 | 0.40e+02 | 0.15e+01 | 0.11e+02 |
| 4         | 0.43e+00 | 0.40e+02 | 0.14e+01 | 0.14e+01 |
| 5         | 0.32e+00 | 0.40e+02 | 0.55e-01 | 0.49e+00 |
| 6         | 0.26e+00 | 0.39e+02 | 0.13e-02 | 0.72e-01 |
| 7         | 0.24e+00 | 0.38e+02 | 0.29e-04 | 0.29e-02 |
| 8         | 0.22e+00 | 0.37e+02 | 0.33e-06 | 0.14e-03 |
| 9         | 0.21e+00 | 0.36e+02 | 0.25e-08 | 0.34e-05 |

**Example 1.4.2.** We start with a simple tridiagonal, diagonally dominant matrix \( A \) with diagonal elements \( a_{jj} = 2.4 + \frac{j}{100} \) for \( j < 200 \), \( a_{200,200} = 2.4 + \frac{200}{1.5} \) and off-diagonal elements 1, of degree \( n = 200 \). The starting vector is \( u = (0.03, 0.03, \cdots , 0.03, 1)^T \) and \( \text{tol} = 1e-10 \). In Figure (1.1) we have plotted the norms of the residual for the largest Ritz pair, computed with the Davidson method and JD method in figure (1.2). In figure (1.1) the dashed and point lines are for the standard preconditioner and GMRES for determining the vector \( t \) respectively. In figure (1.2) the dashed and point lines are for the GMRES and diagonal standard preconditioner for determining the vector \( t \) respectively. We see from figure (1.2) that JD converges faster than Davidson method and we get better results by using GMRES than the standard preconditioner in the both figures.


### Table 1.2: Davidson method

<table>
<thead>
<tr>
<th>Method</th>
<th>Iteration of convergence</th>
<th>error</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard preconditioner</td>
<td>40</td>
<td>1e+03</td>
<td>1.5756 seconds</td>
</tr>
<tr>
<td>GMRES</td>
<td>40</td>
<td>1e-05</td>
<td>1.3104 seconds</td>
</tr>
</tbody>
</table>

### Table 1.3: JD method

<table>
<thead>
<tr>
<th>Method</th>
<th>Iteration of convergence</th>
<th>error</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard preconditioner</td>
<td>10</td>
<td>1e-11</td>
<td>1.014 seconds</td>
</tr>
<tr>
<td>GMRES</td>
<td>4</td>
<td>1e-12</td>
<td>0.3588 seconds</td>
</tr>
</tbody>
</table>

Figure 1.1: Davidson method
Figure 1.2: Jacobi-Davidson method for SEP
Chapter 2

The Quadratic Eigenvalue Problem (QEP)

In this chapter we will discuss a particular class of eigenproblems: polynomial eigenproblems, with focus on the the quadratic case.

2.1 The Generalized Eigenvalue Problem

The generalized eigenvalue problem (GEP) is of the form

\[ Ax = \lambda Bx \]  

(2.1)

where \( A, B \in \mathbb{C}^{n \times n} \), \( \lambda \in \mathbb{C} \) is an eigenvalue and \( 0 \neq x \in \mathbb{C}^n \) is a corresponding eigenvector. It is easy to see that \( \lambda \) is a root of the characteristic equation

\[ \det(A - \lambda B) = 0. \]

**Definition 2.1.1.** The matrix \( A - \lambda B \) is called a matrix pencil. It is denoted by \( (A, B) \). The pencil \( (A, B) \) is called regular if \( \det(A - \lambda B) \neq 0 \). Otherwise, it is called singular.

**The eigenvalues of a regular pencil**

Let \( (A, B) \) be a regular pencil. Then

- Case one: If \( B \) is nonsingular we can write \( B^{-1}Ax = \lambda x \) or \( AB^{-1}y = \lambda y, y = Bx \) which are standard eigenvalue problem. We note that \( AB^{-1} = B(B^{-1}A)B^{-1} \) is similar to \( B^{-1}A \) so the problems share the same eigenvalues.
Case two: If \(B\) is singular, then the degree of \(p(\lambda) = \text{det}(A - \lambda B)\) is less than \(n\). Let it be \(r\). Then zeros of \(p(\lambda)\) are the eigenvalues of the pair \((A, B)\). The convention is to set the \((n - r)\) remaining eigenvalues to be \(\infty\).

**Theorem 2.1.1.** Let \(A, B \in \mathbb{C}^{n \times n}\) and the pair \((A, B)\) be regular. Let \(Q_1\) and \(Q_2\) be unitary matrices such that:

\[
Q_1A Q_2 = T_1 = \begin{pmatrix} t_{11} & \cdots & * \\ & \ddots & \vdots \\ 0 & \cdots & t_{nn} \end{pmatrix}, \quad Q_1B Q_2 = T_2 = \begin{pmatrix} \tilde{t}_{11} & \cdots & * \\ & \ddots & \vdots \\ & & \tilde{t}_{nn} \end{pmatrix}
\]

Then the generalized eigenvalues \(\lambda_i, i = 1, \ldots, n\) of the regular pencil \((A, B)\) are given by:

\[
\lambda_i = \frac{t_{ii}}{\tilde{t}_{ii}}, \text{ if } \tilde{t}_{ii} \neq 0 \\
\lambda_i = \infty, \text{ if } t_{ii} \neq 0, \tilde{t}_{ii} \neq 0
\]

**Proof:** From \(Ax = \lambda Bx\), we have \(Q_1AQ_2Q_2^Hx = \lambda Q_1BQ_2Q_2^Hx\). Define \(y = Q_2^Hx\). Then from the preceding, we have

\[
Q_1AQ_2y = \lambda Q_1BQ_2y
\]

that is, \(T_1y = \lambda T_2y\). Thus, \(\lambda\) is an eigenvalue of the pair \((A, B)\) if and only if it is an eigenvalue of the pair \((T_1, T_2)\). Because \(T_1\) and \(T_2\) are both triangular, \(\text{det}(T_1 - \lambda T_2) = \prod_{i=1}^n (t_{ii} - \lambda \tilde{t}_{ii})\), and vanishes when \(\lambda_i = \frac{t_{ii}}{\tilde{t}_{ii}}\). Also, when \(t_{ii} \neq 0\) but \(\tilde{t}_{ii} = 0\), \(\lambda_i\) can be set to be \(\infty\).

**Theorem 2.1.2.** *(Generalized Schur Decomposition)* If \(A, B \in \mathbb{C}^{n \times n}\) then there exist unitary matrices \(Q\) and \(Z\) such that

\[
Q^H AZ = \tilde{A}, Q^H BZ = \tilde{B}
\]

where \(\tilde{A}\) and \(\tilde{B}\) are upper triangular. For the proof see [2].

### 2.1.1 The QZ Algorithm

The QZ algorithm computes all the eigenvalues of (2.1). There are three steps

**Step one- Reduction to Hessenberg triangular form**

The first step reduces \(A\) to upper Hessenberg form and \(B\) to upper triangular form. An orthogonal matrix \(U\) is found such that, in the 5-by-5 case
2.1. THE GENERALIZED EIGENVALUE PROBLEM

\[ U^T A = \begin{pmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{pmatrix}, \quad U^T B = \begin{pmatrix} * & * & * & * \\ 0 & * & * & * \\ 0 & 0 & * & * \\ 0 & 0 & 0 & * \end{pmatrix} \]

where * represents a nonzero entry. Then zeros are introduced into \( A \) by rotations matrices while keeping the upper triangular structure of \( B \).

Starting with the bottom left corner \( A \) and \( B \) are multiplied by a Givens matrix from the left such that:

\[ A, B \leftarrow (A, B) G_L^T(4, 5) = \begin{pmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ 0 & * & * & * \end{pmatrix}, \quad U^T B = \begin{pmatrix} * & * & * & * \\ 0 & * & * & * \\ 0 & 0 & * & * \\ 0 & 0 & 0 & * \end{pmatrix} \]

A nonzero entry is introduced in \( B \) which is immediately eliminated by multiplying \( A \) and \( B \) with a Givens matrix from the right such that:

\[ A, B \leftarrow (A, B) G_R(4, 5) = \begin{pmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ 0 & * & * & * \end{pmatrix}, \quad U^T B = \begin{pmatrix} * & * & * & * \\ 0 & * & * & * \\ 0 & 0 & * & * \\ 0 & 0 & 0 & * \end{pmatrix} \]

We continue in this manner with pairs of Givens matrices applied to the pencil until we have finally.

\[ A - \lambda B \leftarrow G_L^T(4, 5) G_L^T(3, 4) G_L^T(4, 5) G_L^T(2, 3) G_L^T(3, 4) G_L^T(4, 5) \times (A - \lambda B) G_R(4, 5) G_R(3, 4) G_R(2, 3) G_R(4, 5) G_R(3, 4) G_R(4, 5) \]

Where \( A \) is now upper Hessenberg and \( B \) is upper triangular as required.

**Step two: Deflation**

If \( a_{k+1,k} = 0 \), for some \( k \), then we have

\[ A - \lambda B = \begin{pmatrix} A_{11} - \lambda B_{11} & A_{12} - \lambda B_{12} \\ 0 & A_{22} - \lambda B_{22} \end{pmatrix} \]

and we can solve the two smaller problems \( A_{11} - \lambda B_{11} \) and \( A_{22} - \lambda B_{22} \) independently. If a zero appears on the diagonal of \( B \), in any position, it is possible to introduce a zero in the \((n, n-1)\) position of \( A \) and also chase the zero on \( B \)’s diagonal to the \((n, n)\) position, and thus deflate the problem. This is achieved with pairs of Givens matrices in a similar way as used in step one. After this process we have
CHAPTER 2. THE QUADRATIC EIGENVALUE PROBLEM (QEP)

\[ A - \lambda B = \begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix} - \lambda \begin{pmatrix} B_{11} & B_{12} \\ 0 & B_{22} \end{pmatrix} \]

Where \( A_{11} \) is upper Hessenberg, \( a_{22} \neq 0 \) is a scalar, and \( B_{11} \) is upper triangular. We thus solve \( A_{11} - \lambda B_{11} \) and have an infinite eigenvalue given by

\[ \lambda = \frac{a_{22}}{0}. \]

**Step three - The QZ Step**

The QZ step applies unitary transformations, \( Q \) and \( Z \) to the deflated \( A - \lambda B \); such that

\[ Q^H (A - \lambda B) Z = S - \lambda T \]

where \( S \) and \( T \) are upper triangular. In the real case \( S \) may only be reduced to quasiupper triangular form. That is block upper triangular with 1-by-1 and 2-by-2 blocks. It can be reduced to upper triangular form if \( S \) is allowed to become complex. This is an iterative process. It is derived from considering one step of the standard shifted QR algorithm on the matrix \( AB^{-1} \), which is never formed.

**Definition 2.1.2.** The pair \((A, B)\) with \( A, B \) Hermitian is definite if

\[ \delta(A, B) := \min_{\|z\|=1, z \in \mathbb{C}} \sqrt{(z^*Az)^2 + (z^*Bz)^2} > 0. \]

Immediately, if \( B \) is positive definite then \((A, B)\) is definite.

**2.2 QEP**

In this section we will pay attention to the important class of quadratic eigenproblems, with a small sidestep to higher order polynomial eigenproblems.

**Definition 2.2.1.** A matrix function \( P : \mathbb{C} \longrightarrow \mathbb{C}^{n \times n} \) is called the quadratic matrix pencil if \( P(\lambda) = (\lambda^2M + \lambda C + K) \), where \( M \neq 0 \), \( C \) and \( K \) are given square matrices and \( \overline{\mathbb{C}} := \mathbb{C} \cup \{z\infty \mid z \in \mathbb{C}, \mid z \mid = 1\} \).

**Definition 2.2.2.** A scalar \( \lambda \in \mathbb{C} \) such that \( \det(P(\lambda)) = 0 \) is called an eigenvalue of the quadratic pencil \( P \). We denote by \( \Lambda(P) \) the spectrum of \( P \)

\[ \Lambda(P) = \{\lambda \in \mathbb{C} : \det(P(\lambda)) = 0\}, \]

that is, the set of all eigenvalues of \( P(\lambda) \).
Definition 2.2.3. The nonzero vectors $x$ and $y$ are, respectively, called the right and left eigenvectors, corresponding to $\lambda$ of the quadratic pencil $P(\lambda) = (\lambda^2 M + \lambda C + K)$ if

$$\begin{align*}
(\lambda^2 M + \lambda C + K)x &= 0 \\
y^H(\lambda^2 M + \lambda C + K) &= 0.
\end{align*}$$

The triplet $(\lambda, x, y)$ is called the eigentriplet of $P$.

The quadratic eigenvalue problem (3.4) is the problem of determining all the eigenvalues and the corresponding eigenvectors of the given quadratic pencil $P(\lambda)$. The standard eigenvalue and generalized problems are special cases of QEP. We will assume that $P(\lambda)$ is regular, that is $\det(P(\lambda)) \neq 0$. The QEP has $2n$ eigenvalues (finite or infinite) with up to $2n$ right and $2n$ left eigenvectors, and if there are more than $n$ eigenvectors they do not form a linearly independent set.

Example 2.2.1. Let $P(\lambda)$ be defined by

$$P(\lambda) = \begin{pmatrix}
\lambda + 1 & 6\lambda^2 - 6\lambda & 0 \\
2\lambda & 6\lambda^2 - 7\lambda + 1 & 0 \\
0 & 0 & \lambda^2 + 1
\end{pmatrix}$$

or equivalently by

$$M = \begin{pmatrix}
0 & 6 & 0 \\
0 & 6 & 0 \\
0 & 0 & 1
\end{pmatrix}, \quad C = \begin{pmatrix}
1 & -6 & 0 \\
2 & -7 & 0 \\
0 & 0 & 0
\end{pmatrix}, \quad K = I,$$

is regular because

$$\det(P(\lambda)) = -6\lambda^5 + 11\lambda^4 - 12\lambda^3 + 12\lambda^2 - 6\lambda + 1 = 5 \neq 0.$$ There are six eigenpairs $(\lambda_k, x_k)$, $k = 1, \cdots, 6$ given by

<table>
<thead>
<tr>
<th>$k$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda_k$</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{2}$</td>
<td>1</td>
<td>$i$</td>
<td>$-i$</td>
<td>$\infty$</td>
</tr>
<tr>
<td>$x_k$</td>
<td>$\begin{pmatrix} 1 \ 1 \ 0 \end{pmatrix}$</td>
<td>$\begin{pmatrix} 1 \ 1 \ 0 \end{pmatrix}$</td>
<td>$\begin{pmatrix} 0 \ 1 \ 1 \end{pmatrix}$</td>
<td>$\begin{pmatrix} 0 \ 0 \ 1 \end{pmatrix}$</td>
<td>$\begin{pmatrix} 0 \ 0 \ 1 \end{pmatrix}$</td>
<td>$\begin{pmatrix} 1 \ 0 \ 0 \end{pmatrix}$</td>
</tr>
</tbody>
</table>
Five of the eigenvalues are finite and one of them is infinite. We see that for \( \lambda_1 \neq \lambda_2 \) we have \( x_1 = x_2 \). This example illustrates the fact that if a regular \( P(\lambda) \) has \( 2n \) distinct eigenvalues, then there exists a set of \( n \) linearly independent eigenvectors, which is a nontrivial generalization of standard results for the SEP and the GEP.

**Definition 2.2.4.** The algebraic multiplicity of an eigenvalue \( \lambda_0 \) is the order, \( \alpha \), of the corresponding zero in characteristic polynomial of \( P(\lambda) \) \((\det(P(\lambda)) = \det(M)\lambda^{2n} + \text{lower order terms})\). The geometric multiplicity of \( \lambda_0 \), \( \gamma \), is the dimension of \( \ker(P(\lambda)) \).

- For a simple eigenvalue \( \alpha = \gamma = 1 \).
- For a semi-simple eigenvalue \( \alpha = \gamma \).
- For a defective eigenvalue \( \alpha \neq \gamma \).

We say that \( x_1 \) is a generalized eigenvector corresponding to \( \lambda_0 \) if \( x_1 \) is a solution of the equation

\[
P(\lambda)x_1 = -P'(\lambda_0)x_0
\]

for some eigenvector \( x_0 \), where \( P'(\lambda_0) = 2\lambda_0M + C \). A semi-simple eigenvalue is an eigenvalue for which there is no generalized eigenvector.

We say that \( x_0, \ldots, x_{m-1} \) form a Jordan chain of length \( m \) for \( P(\lambda) \) corresponding to \( \lambda_0 \) if

\[
P(\lambda_0)x_0 = 0,
P(\lambda_0)x_1 + P'(\lambda_0)x_0 = 0,
P(\lambda_0)x_2 + P'(\lambda_0)x_1 + \frac{1}{2}P''(\lambda_0)x_0 = 0,
\vdots
P(\lambda_0)x_{m-1} + P'(\lambda_0)x_{m-2} + \frac{1}{2}P''(\lambda_0)x_{m-3} = 0,
\]

**The Jordan decomposition** of a matrix \( A \in \mathbb{C}^{n \times n} \) is given by:

\[
A = XJX^{-1}
\]

where \( X \) is nonsingular matrix, contains the eigenvectors and generalized eigenvectors of \( A \), \( J = \text{diag}(J_1, \ldots, J_t) \), \( J_k \) is called a Jordan block:

\[
J_k = \begin{pmatrix}
\lambda_k & 1 & \cdots & 0 \\
0 & \lambda_k & \ddots & \vdots \\
\vdots & \ddots & \ddots & 1 \\
0 & \cdots & 0 & \lambda_k
\end{pmatrix} \in \mathbb{C}^{m_k \times m_k}
\]
where \( m_1 + \cdots + m_t = n \). \( m_k \) is the partial multiplicity of \( \lambda_k \). A multiple \( \lambda \) may have several partial multiplicity and the algebraic multiplicity of \( \lambda \) is equal to the sum of its partial multiplicity. The geometric multiplicity of \( \lambda \) is the number of the partial multiplicities.

**Jordan triples and its properties for** \( P(\lambda) \)

We want to generalize Jordan form of a single matrix \( A \). Assume \( M \) is nonsingular and \( J = \text{diag}(J_1, \ldots, J_t) \), where \( J_k \) is Jordan block of size \( m_k \) for \( k = 1, \ldots, t \), \( m_1 + \cdots + m_t = 2n \). Partition \( X \) conformably with \( J \):

\[
X = [X_1, \ldots, X_t] \in \mathbb{C}^{n \times 2n},
\]

where \( X_k \in \mathbb{C}^{n \times m_k} \) for \( k = 1, \ldots, t \). Columns of \( X_k = [x_0^k, \ldots, x_{m_k-1}^k] \) form Jordan chain for \( P(\lambda) \) corresponding to \( \lambda_k \).

Left Jordan chains are obtained from

\[
Y = \begin{pmatrix} X \\ XJ \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ I \end{pmatrix} M^{-1} \in \mathbb{C}^{2n \times n}
\]

\((X, J, Y)\) form a Jordan triplet.

**Example 2.2.2.** \( P(\lambda) = \lambda^2 I + \lambda \left( \begin{array}{cc} 0 & 0 \\ 1 & -1 \end{array} \right) + \left( \begin{array}{cc} 0 & 0 \\ 1 & 0 \end{array} \right) = \left( \begin{array}{cc} \lambda^2 & 0 \\ \lambda + 1 & \lambda(\lambda - 1) \end{array} \right) \)

\[
det(P(\lambda)) = \lambda^3(\lambda - 1) \implies \Lambda(\lambda) = \{0, 1\}.
\]

\( P(0) = \left( \begin{array}{cc} 0 & 0 \\ 1 & 0 \end{array} \right), P(1) = \left( \begin{array}{cc} 1 & 0 \\ 2 & 0 \end{array} \right) \implies x_0 = \left( \begin{array}{c} 0 \\ \alpha \end{array} \right) \) is an eigenvector for both \( \lambda = 0 \) and \( \lambda = 1 \).

\[
P(0)x_1 + P'(0)x_0 = 0 \implies x_1 = \left( \begin{array}{c} \alpha \\ \beta \end{array} \right).
\]

\[
P(0)x_2 + P'(0)x_1 + P''(0)x_0 = 0 \implies x_2 = \left( \begin{array}{c} -2\alpha + \beta \\ \gamma \end{array} \right).
\]

Hence with the choice \( \alpha = 1 \) and \( \beta = \gamma = 0 \),

\[
X = \left( \begin{array}{ccc} 0 & 1 & -2 \\ 0 & 1 & 0 \end{array} \right), J = \left( \begin{array}{ccc} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{array} \right)
\]

If \( \left( \begin{array}{c} X \\ XJ \end{array} \right) \) is nonsingular then \( MXJ^2 + CXJ + KX = 0 \). The \( X \) and \( Y \) satisfy the bi-orthogonality condition:
\[(Y JY) \begin{pmatrix} C & M \\ M & 0 \end{pmatrix} \begin{pmatrix} X \\ XJ \end{pmatrix} = I.\]

\(M, C\) and \(K\) can be expressed in terms of \((X, J, Y)\):
\[
M = (XJY)^{-1}, \begin{pmatrix} K \\ C \end{pmatrix} = -MXJ^2 \begin{pmatrix} X \\ XJ \end{pmatrix}^{-1}.
\]

- Inverse of \(P(\lambda)\):
\[
P(\lambda)^{-1} = X[\lambda I - J]^{-1} \begin{pmatrix} X \\ MXJ \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ I \end{pmatrix}
\]

- If all eigenvalues are semi-simple:
\[
P(\lambda)^{-1} = X(\lambda I - \Lambda)^{-1}Y^* = \sum_{i=1}^{2n} \frac{x_iy_i^*}{\lambda - \lambda_i}.
\]

For more details see [19]

**Definition 2.2.5.** \(P(\lambda)\) is self adjoint if \(P(\lambda) = P(\bar{\lambda})^H\) for all \(\lambda \in \mathbb{C}\).

\(P(\lambda)\) is self adjoint \(\iff\) \(M, C\) and \(K\) hermitian.

**Spectral properties**
- Eigenvalues are real or arise in pairs \((\lambda, \bar{\lambda})\).
- \(P(\lambda) = 0 \iff x^*P(\bar{\lambda})\), i.e., \(x\) is a right eigenvector of \(\lambda\) and a left eigenvector of \(\bar{\lambda}\).
- \(M, C\) and \(K\) are real symmetric \(\iff\) sets of left and right eigenvectors coincide.

**Note:** For any eigenpair \((\lambda, x)\)
\[
\lambda^2 x^* M x + \lambda x^* C x + x^* K x = 0,
\]
with solutions \(\lambda = \frac{-x^* C x \pm \sqrt{(x^* C x)^2 - 4(x^* M x)(x^* K x)}}{2x^* M x}\).

**Definition 2.2.6.** \(P(\lambda)\) is called hyperbolic if \(M\) is hermitian positive definite, \(C\) and \(K\) are hermitian and
\[
(x^* C x)^2 > 4(x^* M x)(x^* K x) \text{ for all } x \neq 0.
\]
2.2. QEP

Hyperbolic QEP has real eigenvalues with
\[ \lambda_1 \geq \cdots \geq \lambda_n \geq \lambda_{n+1} \geq \cdots \geq \lambda_{2n}. \]

**Definition 2.2.7.** \( P(\lambda) \) is elliptic if \( M \) is hermitian positive definite, \( C \) and \( K \) are hermitian and
\[ (x^*Cx)^2 < 4(x^*Mx)(x^*Kx) \text{ for all } x \neq 0. \]
Elliptic QEP has nonreal eigenvalues, and, necessarily, \( K \) is positive definite.

We introduced two classes of self adjoint \( P(\lambda) \) corresponding to \( \Lambda(P) \subset \mathbb{R} \) and \( \Lambda(P) \cap \mathbb{R} = \emptyset \)

**Definition 2.2.8.** The \( P(\lambda) \) is overdamped if it is hyperbolic with \( C \) hermitian positive definite and \( K \) hermitian positive semi-definite.

In this case, the condition for hyperbolicity
\[ \min_{\|x\|=1} [(x^*Cx)^2 - 4(x^*Mx)(x^*Kx)] > 0 \]
is called overdamping condition.

Properties of overdamped problems:

- Eigenvalues are real and nonpositive \( \implies \) the corresponding system is stable.
- Gaps between \( n \) largest eigenvalues and \( n \) smallest eigenvalues.
- There are \( n \) linearly independent eigenvectors associated with \( n \) largest eigenvalues and likewise for the \( n \) smallest eigenvalues.

Applications of QEP

From, e.g., vibration analysis of structural systems,
\[ M\ddot{q}(t) + C\dot{q}(t) + Kq(t) = f(t). \]
where \( q(t) \) is an \( n \)th-order vector and \( M, C \) and \( K \) are \( n \times n \) matrices.
- As we know from analysis, the general solution of the homogeneous equation is:
\[ q(t) = \sum_{i=1}^{2} \alpha_i e^{\lambda_i t}. \]
• Since each term \( x_i = \alpha_i e^{\lambda_i t} \) is also solution, we have to find a \( \lambda_i \) which satisfies the QEP’s equation:

\[
(\lambda_i^2 M + \lambda_i C + K)x_i = 0.
\]

**Example 2.2.3. The Millenium Bridge**

With help of the QEP and model of the bridge, the engineers found the smallest resonance frequency to be 0.9 Hz. This is approximately the frequency of the footsteps of the pedestrians walking on the bridge.

• In the Finite Element Method, we discretize a continuous structure like a bridge.

• Each of the mass points is linked to a few others with help of springs and dampers.

• For each mass point, we write the force equation. In matrix notation, this gives

\[
M\ddot{q}(t) + C\dot{q}(t) + Kq(t) = f(t).
\]

where \( M \) mass matrix, \( C \) damping matrix and \( K \) stiffness matrix.

In figure (2.1) all \( \text{Re}(\lambda) \leq 0 \) therefore the bridge is stable. In figure (2.2) all the eigenvalues are real and nonpositive, so the bridge is totally safe. Even if many people walk on it, it won’t wobble.
Mass-spring system

Vibration of the system governed by

\[ M\ddot{q}(t) + C\dot{q}(t) + Kq(t) = f(t). \]

Take \( n = 50, \quad \kappa_i = \kappa, \quad i \neq 1, n, \quad \kappa_1 = \kappa_n = 2\kappa \)
\( \tau_i = \tau, \quad i \neq 1, n, \quad \tau_1 = \tau_n = 2\tau, \quad m_i \equiv 1. \)

\[ M = I > 0, \quad C = \tau \text{ tridiag}(-1, 3, -1) > 0, \]
\[ K = \kappa \text{ tridiag}(-1, 3, -1) > 0. \]

Consider two cases:

case 1 : \( \kappa = 5, \quad \tau = 3, \) (non overdamped).

case 2 : \( \kappa = 5, \quad \tau = 10, \) (overdamped).
Figure 2.1: Eigenvalues of the QEP for the non-overdamped mass-spring system with $n=50$

Figure 2.2: Eigenvalues distributions of the QEP for the overdamped mass-spring system with $n=50$
2.2. QEP

2.2.1 The solution of QEP

The most widely used approach for solving the QEP (even the polynomial eigenvalue problem \( P(\lambda) x = (\sum_{i=0}^{m} \lambda^i A_i) x = 0 \) in \( n \times n \) matrices \( A_i \)) is to linearize and produce a larger order pencil \( L(\lambda) = \lambda X + Y \). This generalized eigenproblem is usually solved with the method QZ for small to medium size problems or a projection method for large sparse problems, i.e., JD method.

The concept of the Linearization

Reduction of 2nd order to differential equation to 1st order

\[
\begin{align*}
M\ddot{q}(t) + C\dot{q}(t) + Kq(t) &= f(t), \\
q(t), f(t) &\in \mathbb{C}^n.
\end{align*}
\]

Define \( p_1 = q, p_2 = \dot{p}_1 \). Then

\[
\begin{align*}
M\dot{p}_2 + Cp_2(t) + Kp_1 &= f(t) \\
\dot{p}_1 &= p_2.
\end{align*}
\]

Reduction of QEP to GEP:

\[
(\lambda^2 M + \lambda C + K)x = 0.
\]

Define \( y_1 = x, y_2 = \lambda x \). Then

\[
(\lambda \begin{pmatrix} I & 0 \\ 0 & M \end{pmatrix} + \begin{pmatrix} 0 & -I \\ K & C \end{pmatrix})y = 0
\]

Definition 2.2.9. A matrix polynomial \( E(\lambda) \) is unimodular if its determinant is a nonzero constant, independent of \( \lambda \).

Definition 2.2.10. (Linearization) Let \( P(\lambda) \) be an \( n \times n \) matrix polynomial of degree \( k \) with \( k \geq 1 \). A pencil \( L(\lambda) = \lambda X + Y \) with \( X, Y \in \mathbb{C}^{kn \times kn} \) is called linearization of \( P(\lambda) \) if there exist unimodular matrix polynomials \( E(\lambda), F(\lambda) \in \mathbb{C}^{kn \times kn} \) such that

\[
F(\lambda)L(\lambda)E(\lambda) = \begin{pmatrix} P(\lambda) & 0 \\ 0 & I_{(k-1)n} \end{pmatrix}
\]

Remark 2.2.1.

- For \( k = 2 \) gives us the linearization for matrix polynomial of degree 2 (i.e., QEP).
- The linearization is not unique.

Most of the linearizations are used of the first or second companion form

- First companion form:
CHAPTER 2. THE QUADRATIC EIGENVALUE PROBLEM (QEP)

\[
\begin{pmatrix}
0 & W \\
-K & -C
\end{pmatrix} - \lambda \begin{pmatrix}
W & 0 \\
0 & M
\end{pmatrix}
\]

- Second companion form:

\[
\begin{pmatrix}
-K & 0 \\
0 & W
\end{pmatrix} - \lambda \begin{pmatrix}
C & M \\
W & 0
\end{pmatrix}
\]

\(W \in \mathbb{C}^{n \times n}\) can be any nonsingular matrix.

We will give some informations about symmetric and Skew hermitian/hermitian linearization

1. Symmetric Linearization

- \(M \succ 0\) (hermitian positive definite), \(W = I\) in first companion form

\[
\begin{pmatrix}
0 & I \\
-K & -C
\end{pmatrix} - \lambda \begin{pmatrix}
I & 0 \\
0 & M
\end{pmatrix}
\]

- \(K \succ 0\) (hermitian positive definite), \(W = I\) in the second companion form

\[
\begin{pmatrix}
C & M \\
I & 0
\end{pmatrix} - \frac{1}{\lambda} \begin{pmatrix}
-K & 0 \\
0 & I
\end{pmatrix}
\]

- \(W = -K\) is first companion form

\[
\begin{pmatrix}
0 & K \\
K & C
\end{pmatrix} - \lambda \begin{pmatrix}
K & 0 \\
0 & -M
\end{pmatrix}
\]

- \(W = M\) in second companion form

\[
\begin{pmatrix}
-K & 0 \\
0 & M
\end{pmatrix} - \lambda \begin{pmatrix}
C & M \\
M & 0
\end{pmatrix}
\]

2. Skew hermitian/hermitian linearization

- \(N = -K\) in first companion and block column swap

\[
\begin{pmatrix}
K & 0 \\
C & K
\end{pmatrix} - \lambda \begin{pmatrix}
0 & K \\
-M & 0
\end{pmatrix}
\]
2.2. QEP

- \( N = -M \) in second companion form and block column swap
  \[
  \begin{pmatrix}
  0 & -K \\
  M & 0 \\
  \end{pmatrix} - \lambda \begin{pmatrix}
  M & C \\
  0 & M \\
  \end{pmatrix}
  \]

Remark 2.2.2.

- We can solve the QEP by JD method and we will discuss JD method for higher order polynomial eigenvalue problems in the next chapter.

- To enable us to deal more easily with both finite or infinite eigenvalues we will often write the QEP in homogeneous form, as
  \[ P(\alpha, \beta) = \alpha^2 M + \alpha \beta C + \beta^2 K. \]

The linearization \( L(\lambda) = \lambda X + Y \) can also be written in homogeneous form, as
  \[ L(\alpha, \beta) = \alpha X + \beta Y. \]

The homogeneous form of QEP can be derived by substituting \( \lambda = \frac{\alpha}{\beta} \) into
  \[ P(\lambda) = \lambda^2 M + \lambda C + K \] yielding
  \[ P(\alpha, \beta) = (\frac{\alpha}{\beta})^2 M + (\frac{\alpha}{\beta}) C + K. \]  \hspace{1cm} (2.4)

On multiplying (3.5) by \( \beta^2 \) we obtain the homogeneous form \( P(\alpha, \beta) = \alpha^2 M + \alpha \beta C + \beta^2 K \). We associate an eigenvalue \( \lambda = \frac{\alpha}{\beta} \) with the pair \((\alpha, \beta)\). The representation of \( \lambda \) as \( \frac{\alpha}{\beta} \) is not unique, since for \( \gamma \neq 0 \) we have that \( (\gamma \alpha, \gamma \beta) \) yields \( \lambda = \frac{(\gamma \alpha)}{(\gamma \beta)} = \frac{\alpha}{\beta} = \lambda \). We can express \( \lambda \) uniquely using normalized form, by requiring that \( |\alpha|^2 + |\beta|^2 = 1 \). In normalized form \((0, 1)\) represents a zero eigenvalue and \((1, 0)\) an infinite eigenvalue.

Example 2.2.4. We consider QEP \((\lambda^2 M + \lambda C + K)x = 0\) where \( M, C, K \in \mathbb{R}^{n \times n} \) are symmetric, with \( n = 200 \). In this example \( C = e_{100}e_{100}^T \) and \( M \) and \( K \) are positive definite. Figures (2.3) and (2.4) illustrate the sensitivity of the linearization process. We plotted in the complex plane the eigenvalues of QEP that we first solved by the companion linearization in Figure (2.3). After the linearization, we used the QZ algorithm. We see in Figure (2.3) that some eigenvalues computed using the companion linearization have a positive part whereas those computed with the symmetric linearization have a negative or 0 real part. Thus, in this case the symmetric linearization performs better than the companion linearization.
Figure 2.3: Spectrum computed with the companion linearization

Figure 2.4: Spectrum computed with the symmetric linearization
Chapter 3

Higher Order Polynomial Eigenvalue Problems (PEP)

We know that the polynomial eigenvalue problem (PEP) is defined by

\[ P(\lambda)x = \left( \sum_{i=0}^{m} \lambda^i A_i \right)x = (A_0 + \lambda A_1 + \cdots + \lambda^m A_m)x = 0, \quad (3.1) \]

where \( A_i \in \mathbb{C}^{n \times n} \).

Remark 3.0.3. All definitions for QEP also satisfy for PEP.

We can reduce (3.1) to GEP of size \( mn \times mn \) where \( m \geq 2 \) is degree of PEP as follows:

\[
\begin{pmatrix}
\lambda & A_1 & A_2 & \cdots & A_m \\
I & & & & \\
& \ddots & & & \\
& & I & & \\
& & & I & \\
\end{pmatrix}
+ \begin{pmatrix}
A_0 & -I & & & \\
& -I & & & \\
& & \ddots & & \\
& & & -I & \\
& & & & \lambda^{m-1}I \\
\end{pmatrix}
\begin{pmatrix}
x \\
x \\
\lambda x \\
\vdots \\
\lambda^{m-1}x \\
\end{pmatrix}
= 0
\]

Another linearization is

\[
\begin{pmatrix}
\lambda & A_1 & A_2 & \cdots & A_m \\
A_2 & & & & \\
& \ddots & & & \\
& & A_m & & \\
& & & I & \\
\end{pmatrix}
+ \begin{pmatrix}
A_0 & -A_2 & & & \\
& -A_2 & & & \\
& & \ddots & & \\
& & & -A_m & \\
& & & & \lambda^{m-1}I \\
\end{pmatrix}
\begin{pmatrix}
x \\
x \\
\lambda x \\
\vdots \\
\lambda^{m-1}x \\
\end{pmatrix}
= 0
\]

Both are equivalent, if \( A_2, \cdots, A_m \) are regular.
3.1 JD method for PEP

We want to generalize the JD method for the PEP [15, 17, 18].

- Given an approximate eigenvalue $\theta \approx \lambda$ and an approximate eigenvector $u \approx x$, the residual vector $r$ is defined by

$$r(\theta, u) := P(\theta)u = \left(\sum_{i=0}^{m} \theta^i A_i\right)u$$

Then the Rayleigh-Ritz approach for the PEP imposes the Galerkin condition $P(\theta)u \perp K$, where $K = \text{span}\{v_1, \cdots, v_k\}$ and $v_1, \cdots, v_k$ are a set of orthonormal basis vectors of $K$. With $u = V_k s$ ($V_k = [v_1, \cdots, v_k]$ and $s \in \mathbb{C}^k$) leading to the projected polynomial eigenproblem

$$(V^H A_0 V + \theta V^H A_1 V + \cdots + \theta^m V^H A_m V)s = 0$$

which, as a low-dimensional problem, may for instance be solved using techniques employing a companion matrix. We consider a Galerkin condition $P(\theta)u \perp K$ where $K$ is a suitable choosen test space. For every test vector $v \in K$, this implies $P(\theta)u \perp v$.

**Note:** that $(\theta, u)$ is a Ritz pair.

- We are interested in an update $t \perp u$ such that

$$P(\lambda)(u + t) = 0. \quad (3.2)$$

$\lambda = \theta + \eta$ is an exact eigenvalue of $P$ associated to $x = u + t$. The last equation is a nonlinear equation that cannot be solved directly, we can apply a newton step on the system

$$f(x, \lambda) := \begin{pmatrix} P(\lambda)x \\ x^H x - 1 \end{pmatrix} \quad (3.3)$$

where $\|x\| = 1$.

- Let we have a solution $(x_0, \lambda_0) \in \mathbb{C}^{n+1}$ then we get the correction vector $(x_1, \lambda_1) - (x_0, \lambda_0) = (\eta, t) \in \mathbb{C}^{n+1}$ from
3.1. JD METHOD FOR PEP

\[ f'(x_0, \lambda_0) \begin{pmatrix} \eta \\ t \end{pmatrix} = -f(x_0, \lambda_0). \]

- We have

\[ f'(\theta, u) \begin{pmatrix} \eta \\ t \end{pmatrix} = -f(\theta, u) \iff \begin{pmatrix} P'(\theta) u & P(\theta) \\ 0 & 2u^H \end{pmatrix} \begin{pmatrix} \eta \\ t \end{pmatrix} = \begin{pmatrix} -r \\ 0 \end{pmatrix} \]

where \( r = P(\theta)u \) and \( P'(\lambda) = \frac{\partial}{\partial \lambda} P(\lambda) \).

- The first block line is multiplied by \( u \) from the left

\[ \eta u^H P'(\theta) u + u^H P(\theta) t = -u^H r = 0 \iff \eta = \frac{u^H P(\theta) u}{u^H P'(\theta) u}. \quad (3.4) \]

where \( u^H r = u^H P(\theta) u = s^H V^H P(\theta) Vs = 0 \).

- The second block line demands

\[ t \perp u \iff \left( I - \frac{uu^H}{u^Hu} \right) t = t \] \quad (3.5)

an orthogonal projection onto \( u^\perp \).

- equation (3.4) and (3.5) lead to correction equation

\[ \left( I - \frac{P'(\theta) uu^H}{u^H P'(\theta) u} \right) P(\theta) \left( I - \frac{uu^H}{u^Hu} \right) t = -r. \]

- Solve approximately the correction equation and we want to add this solution to \( V \) such that \( V_{\text{new}} := \text{orth}[V_{\text{old}}, v] \), where \( v = \frac{t}{\| t \|} \).

Remark 3.1.1. The following algorithm is JD method for polynomial eigenvalue problem and we can apply this method for generalization eigenvalue problem (GEP) and quadratic eigenvalue problem (QEP).
Algorithm 7 J-D method for polynomial eigenvalue problem

1: Start: choose an $n \times k$ orthonormal matrix $V$.
2: for $i = 0$ to $m$ do
3:   Compute $W_i = A_i V$ and $M_i = V^H W_i$.
4:   Compute the eigenpair $(\theta, s)$ of

$$P_i(\theta)s = \left( \sum_{i=0}^{m} \theta^i M_i \right)s = 0, M_i \in \mathbb{C}^{n \times n}.$$

5: Select the desired eigenpair $(\theta, s)$ with $\|s\| = 1$.
6: Compute $u = Vs$, $\omega = P'(\theta)u, r = P(\theta)u$.
7: if $\|r\| < \epsilon$ then
8:   $\lambda = \theta$
9:   $x = u$
10: STOP
11: end if
12: Solve correction equation (approximately)

$$\left( I - \frac{uH}{uu^H} \right) P_i(\theta) \left( I - uu^H \right) t = -r$$

13: Orthogonalize $t$ against $V$, $v = \frac{t}{\|t\|}$.
14: for $i = 0$ to $m$ do
15:   Compute $w_i = A_i v$
16:   $$M_i = \begin{pmatrix} M_i & V_i W_i \\ v_i W_i & v_i W_i \end{pmatrix}, W_i = [W_i, w_i]$$
17: end for
18: Expand $V = [V, v]$
3.2 Solving the correction equation

We can solve the correction equation by using GMRES (see (1.4.2) and (A.2.2)), inverse projections, one step approximation, ILU decomposition and BiCG method. In this subsection we want to discuss the last four methods.

1. Inverse Projection:

We want to compute \( t \) from the correction equation:

\[
\left( I - \frac{P'(\theta)uu^H}{u^H P'(\theta)u} \right) P(\theta) \left( I - \frac{uu^H}{u^H u} \right) t = r
\]

By multiplying both sides from the left by the projection matrix \( P = \left( I - \frac{P'(\theta)uu^H}{u^H P'(\theta)u} \right) \) (note that \( P^2 = P \)) so that we get

\[
\left( I - \frac{P'(\theta)uu^H}{u^H P'(\theta)u} \right) P(\theta) \left( I - \frac{uu^H}{u^H u} \right) t = \left( I - \frac{P'(\theta)uu^H}{u^H P'(\theta)u} \right) r. \tag{3.6}
\]

We know that the solution of the system \( Mx = Pb \) is equivalent to the solution of the system \( PMx = Pb \) for \( M \in \mathbb{C}^{n \times n} \) and \( b \in \mathbb{C}^{n} \), so that the equation (3.6) can be rewritten as follows:

\[
P(\theta) \left( I - \frac{uu^H}{u^H u} \right) t = \left( I - \frac{P'(\theta)uu^H}{u^H P'(\theta)u} \right) r.
\]

Note: If \( P(\theta) \) is regular then we can compute the inverse \( P(\theta)^{-1} \).

We get

\[
\left( I - \frac{uu^H}{u^H u} \right) t = P(\theta)^{-1} \left( I - \frac{P'(\theta)uu^H}{u^H P'(\theta)u} \right) r.
\]

and the trick with inverse projection is done again in order to get the solution \( t \)

\[
t = \left( I - \frac{uu^H}{u^H u} \right) P(\theta)^{-1} \left( I - \frac{P'(\theta)uu^H}{u^H P'(\theta)u} \right) r. \tag{3.7}
\]

**Definition 3.2.1.** Let \( P : \mathbb{C}^{n} \rightarrow \mathbb{C}^{n} \) be an operator. We call \( Q : \mathbb{C}^{n} \rightarrow \mathbb{C}^{n} \) approximative inverse of \( P \), if \( Qb \approx x, \forall x, b \in \mathbb{C}^{n} \) for \( Px = b \).

We can rewrite the equation (3.7) as follows
CHAPTER 3. HIGHER ORDER POLYNOMIAL EIGENVALUE PROBLEMS (PEP)

\[ t = \left( I - \frac{uu^H}{u^Hu} \right) Q(\theta) \left( I - \frac{P'(\theta)uu^H}{u^H P'(\theta)u} \right) r. \]

2. One step approximation

We want to solve the correction by one step approximation [15].

\[
\left( I - \frac{P'(\theta)uu^H}{u^H P'(\theta)u} \right) P(\theta) \left( I - \frac{uu^H}{u^Hu} \right) t = -r
\]

\[\iff\]

\[
\left( I - \frac{P'(\theta)uu^H}{u^H P'(\theta)u} \right) P(\theta)t = -r, \text{ since } u^H t = 0
\]

By multiplying both sides from the left with inverse \(Q(\theta)\)

\((Q(\theta)P(\theta) \approx I)\) we get

\[ Q(\theta)P(\theta)t - Q(\theta)\frac{P'(\theta)uu^H}{u^H P'(\theta)u} P(\theta)t = -Q(\theta)r \]

\[\iff\]

\[ t = Q(\theta)P'(\theta)u \frac{u^H P'(\theta)u}{u^H P'(\theta)u} - Q(\theta)r, \text{ since } u^H t = 0, Q(\theta)P(\theta) \approx I \]

\[ t = Q(\theta)P'(\theta)u\alpha - Q(\theta)r, \text{ where } \alpha = \frac{u^H P'(\theta)t}{u^H P'(\theta)u} \]

Now \(\alpha\) is chosen to make \(t\) orthogonal to \(u\)

\[ 0 = u^H t = \alpha Q(\theta)P'(\theta)u - u^H Q(\theta)r \]

\[\implies\]

\[ t = \alpha Q(\theta)P'(\theta)u - Q(\theta)r, \text{ where } \alpha = \frac{u^H Q(\theta)r}{u^H Q(\theta)P'(\theta)u}. \]

Remark 3.2.1.

- The inverse Projection and one step approximation are not good for solving the correction equation since \(P^{-1}(\theta)\) generally is a dense matrix although \(P(\theta)\) is sparse and \(P(\theta)\) gets singular when \(\theta\) converges close to an eigenvalue. This is the same case with inverse power method (RQ with shift).

- We can solve the correction equation by incomplete lower upper decomposition (ILU) which keeps sparseness.

- The LU decomposition is applicable i.e., \(z = Q(\theta)y\) is replaced by \(P(\theta)z = y\) with the unknown vector \(z\).
3. Incomplete lower upper decomposition (ILU):

We call a factorization incomplete if during the factorization process certain fill elements, nonzero elements in the factorization in positions where the original matrix had a zero, have been ignored. For more details about ILU see chapter 11 of [12].

Let $A$ be a matrix. A general ILU factorization process computes a lower triangular matrix $L$ and an upper triangular matrix $U$, so the residual matrix $R = LU - A$ satisfies certain constraints, such as having zero entries in some locations. We want to describe a general ILU preconditioner geared toward M-matrices.

**Definition 3.2.2.** The nonsingular matrix $A$ is an M-matrix if

$$a_{ij} \leq 0 \text{ for } i \neq j \text{ and } A^{-1} \geq 0.$$ 

**Note:** It can be shown that they exist for M-matrices.

**Lemma 3.2.1.** Let a matrix $A$ be given such that:

(a) $a_{ij} \leq 0$ for $i \neq j$.
(b) $A$ is nonsingular.
(c) $A^{-1} \geq 0$. Then:
(d) $a_{ii} > 0$ for $i = 1, \cdots, n$ i.e., $A$ is M-matrix.
(e) $\rho(B) < 1$ where $B = I - D^{-1}A$.

**Theorem 3.2.1.** Let $A$ be M-matrix and let $A_1$ be the matrix obtained from the first step of Gaussian elimination. Then $A_1$ is an M-matrix.

We define a set $S$ such that $S = \{(i,j) \mid a_{ij} \neq 0\} \subset S_n = \{(i,j) \mid i \neq j, 1 \leq i,j \leq n\}$. We formulate the theorem that guarantees the existences of incomplete factorization for the M-matrix $A$.

**Theorem 3.2.2.** Let $A = (a_{ij})$ be an $n \times n$ M-matrix, then there exists for every $S \subset S_n$ a lower matrix $\tilde{L} = (l_{ij})$, with $l_{ii} = 1$, an upper triangular matrix $\tilde{U} = (u_{ij})$, and a matrix $R = (r_{ij})$ with

- $l_{ij} = 0$, $u_{ij} = 0$, if $(i, j) \in S$.
- $r_{ij} = 0$ if $(i, j) \notin S$.

The factors $\tilde{U}$ and $\tilde{L}$ are uniquely defined by $S$. 
Algorithm 8 ILU

for \( k = 1 \) to \( n - 1 \) do
  \( d = 1/a_{kk} \)
  for \( i = k + 1 \) to \( n \) do
    if \((i, k) \in S\) then
      \( e = da_{ik} \)
      \( a_{ik} = e \)
    end if
  end for
end for

Remark 3.2.2. A position that is zero in \( A \) but not so in an exact factorization is called a fill position, and if it is outside \( S \), the fill there is said to be discarded. Often, \( S \) is chosen to coincide with the rest of nonzero positions in \( A \), discarding all fill. This factorization type is called the incomplete LU factorization of level zero ILU(0).

4. The Bi-conjugate Gradient method (BiCG):

The conjugate gradient method (CG) is not suitable for non-hermitian systems because the residual vectors cannot be made orthogonal with short recurrences [4]. The Bi-conjugate gradient method takes another approach, replacing the orthogonal sequence of residuals by two mutually orthogonal sequences, at the price of no longer providing a minimization.

The update relations for residuals in the CG are augmented in the BiCG by relations that are similar but based on \( B^H \) instead of \( B \). Thus we update two sequences of residuals

\[
r_k = r_{k-1} - \alpha_{k-1}Bp_{k-1}, \quad \tilde{r}_k = \tilde{r}_{k-1} - \alpha_{k-1}B^H\tilde{p}_{k-1}
\]

and two sequences of search directions

\[
p_k = r_k + \beta_{k-1}p_{k-1}, \quad \tilde{p}_k = \tilde{r}_k + \beta_{k-1}\tilde{p}_{k-1}.
\]

The choices
3.3. NUMERICAL EXAMPLE

\[ \alpha_{k-1} = \left\langle \tilde{r}_{k-1}, r_{k-1} \right\rangle / \left\langle p_{k-1}, Bp_{k-1} \right\rangle, \quad \beta_{k-1} = \left\langle \tilde{r}_{k-1}, r_{k-1} \right\rangle / \left\langle \tilde{r}_{k-1}, \tilde{r}_{k-1} \right\rangle, \]

ensure the bi-orthogonality relations

\[ \tilde{r}_i^H r_j = \tilde{p}_i^H Bp_j = 0 \text{ if } i \neq j. \]

Algorithm 9 biCG(\(z, b, B, \varepsilon, k_{\text{max}}\)) Algorithm

\[ r = b - Bz, \quad \tilde{r} = r, \quad \rho_0 = \tilde{r}^H r = 0, \quad \tilde{p} = p = 0, \quad k = 0. \]

while \( \| r \|_2 > \varepsilon \| b \|_2 \) and \( k < k_{\text{max}} \) do

\[ k = k + 1 \]
\[ \rho_k = \tilde{r}^H r, \quad \beta_{k-1} = \frac{\rho_k}{\rho_{k-1}} = \frac{\left\langle \tilde{r}_k, r_k \right\rangle}{\left\langle \tilde{r}_{k-1}, r_{k-1} \right\rangle}, \]
\[ p = r + \beta p, \quad \tilde{p} = \tilde{r} + \beta \tilde{p}. \text{ (update direction vector)} \]
\[ v = Bp. \]
\[ \alpha_{k-1} = \frac{\left\langle \tilde{r}_{k-1}, r_{k-1} \right\rangle}{\left\langle p_{k-1}, Bp_{k-1} \right\rangle}, \]
\[ z = z + \alpha p. \text{ (update iterate)} \]
\[ r = r - \alpha v, \quad \tilde{r} = \tilde{r} - \alpha B^H \tilde{p}. \text{ (update residual)} \]

end while

Remark 3.2.3. All examples in this thesis are tested by MATLAB 7.0.

3.3 Numerical example

In this example we want to study the cubic eigenvalue problem of the form

\[ P(\lambda) = \lambda^3 M_3 + \lambda^2 M_2 + \lambda M_1 + M_0, \quad (3.8) \]

Where \( M_3, M_2, M_1 \) and \( M_0 \) are complex matrix of size n=64. We are interest to find the eigenvalue \( \lambda \) with \( \text{Im}(\lambda) \) maximal, where \( \lambda \) belongs to the spectrum of \( P(\lambda) \) and the corresponding eigenvector.

Notations

The equation (3.8) is the eigenvalue problem for the potential trouble (in laser plasma physics) such that

\[ \left( \lambda^3 f_3(\theta) + \lambda^2 f_2(\theta) + \lambda (f_1(\theta) + g_1(\theta) \frac{\partial^2}{\partial \theta^2}) + f_0 + g_0 \frac{\partial^2}{\partial \theta^2} \right) \Phi(\theta) = 0. \]

Where \( f_j, g_j \) are \( 2\pi \) periodic function from \( \mathbb{C} \). \( \overrightarrow{f} \) is a discrete representation on grid \( \theta \), \( \theta_0 < \theta_1 < \cdots < \theta_N = \theta_0 + 2\pi \). After discretization on the grid
we get \( f_j, g_j \rightarrow \text{Diag}(f_j \overrightarrow{\theta}) \), where \( f(\overrightarrow{\theta}) = \left( \begin{array}{c} f(\theta_1) \\ f(\theta_2) \\ \vdots \\ f(\theta_N) \end{array} \right) \) and the second derivative leads to periodic band matrix (by finite difference) such that

\[
\frac{\partial^2}{\partial \theta^2} \rightarrow \left( \begin{array}{cccc} * & * & * & \cdot \\ * & * & \cdot & * \\ \cdot & \cdot & * & * \\ * & * & * & * \end{array} \right)
\]

In figure (3.3) we solved the correction equation of our problem by four methods GMRES (directly applied to correction equation), inverse projection, one step approximation and BiCG method. We get faster convergence by using BiCG method than other methods. The GMRES converges better than one step approximation and inverse projection. The initial vector \( v = \text{ones}(\text{length}(M2), 1), \text{tol} = 0.000009 \) and the number of the iterations is 100. BiCG converges at iteration 5, GMRES converges at iteration 7, inverse projection converges at iteration 24 and one step approximation converges at iteration 30.

Table 3.1: Comparison between the time of convergence for the last four methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BiCG</td>
<td>0.3432</td>
</tr>
<tr>
<td>GMRES</td>
<td>0.4212</td>
</tr>
<tr>
<td>One step approximation</td>
<td>3.666</td>
</tr>
<tr>
<td>Inverse projection</td>
<td>1.9812</td>
</tr>
</tbody>
</table>

In figure (3.4) we solved the correction equation by three methods ILU, ILU(0) and LU we see that LU converges faster than ILU and ILU(0) (it converges at the iteration 5) and we did not need much time to compute the results like the ILU and ILU(0). The ILU method converges at the iteration 55 and the ILU(0) method converges at the iteration 64 (since the dimension of search subspace is 64). ILU(0) is not good for this example.
Table 3.2: Comparison between ILU, ILU(0) and LU

<table>
<thead>
<tr>
<th>Method</th>
<th>Iteration of convergence</th>
<th>error</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>LU</td>
<td>5</td>
<td>1e-08</td>
<td>0.1248 seconds</td>
</tr>
<tr>
<td>ILU</td>
<td>55</td>
<td>1e-05</td>
<td>448.3001 seconds</td>
</tr>
<tr>
<td>ILU(0)</td>
<td>64</td>
<td>0.01e-00</td>
<td>54.4911 seconds</td>
</tr>
</tbody>
</table>

Figure 3.1: Spectrum of the equation (3.8)
Figure 3.2: the eigenvectors for $\lambda = -0.1677 + 0.3323i$

Figure 3.3: JD method for the equation (3.8)
3.3. NUMERICAL EXAMPLE

Figure 3.4: comparison between ILU, ILU(0) and LU

Figure 3.5: JD method for the equation (3.8) when n=128
CHAPTER 3. HIGHER ORDER POLYNOMIAL EIGENVALUE PROBLEMS (PEP)
Appendix A

A.1 A list of basic iterative methods

**Definition A.1.1.** The determinant of a block upper triangular matrix is equal to the product of the determinant of the individual diagonal blocks. Then if we seek the eigenvalues of a block upper triangular matrix $U$, we have

$$\det(A - \lambda I) = \prod_i \det(U_{ii} - \lambda I).$$

Thus the problem can split into smaller subproblems which can be solved independently. We call this process *deflation*.

**QR Algorithm**

This algorithm computes all the eigenvalues of the standard eigenvalue problem $Ax = \lambda x$. $A$ is unitary transformed to upper Hessenberg $H$. The $QR$ algorithm is applied on $H$, which has the same eigenvalue of $A$. Within the $QR$ algorithm, $H$ remains upper Hessenberg.

$H$ is upper Hessenberg and if $h_{n,n-1}$ is zeroed, one eigenvalue is in the lower right corner and the problem can be reduced to $H := (h_{ij})$, $i,j = 1, \ldots, n - 1$. This is achieved applying the Wilkinson shift: the two eigenvalues of the $2 \times 2$ block in the lower right corner

$$\begin{pmatrix} h_{n-1,n-1} & h_{n-1,n} \\ h_{n,n-1} & h_{n,n} \end{pmatrix}$$

are calculated analytically and the eigenvalue closer to $h_{n,n}$ is chosen for shift.

**Definition A.1.2.** For a regular matrix $A \in \mathbb{C}^{n \times n}$ the LU decomposition

$$LU = A$$
consists of a lower triangular matrix $L$ and an upper triangular matrix $U$. The diagonal of $L$ is always one. The factors $L$ and $U$ are obtained by Gaussian elimination, where necessary with row interchange by a permutation matrix $P$:

$$LU = PA$$

**Remark A.1.1.** Linear equation systems $Ax = b$ can be solved with backward and forward elimination by the LU decomposition:

$$LUx = Ax = b \iff Ly = b \text{ and } Ux = y$$

**Algorithm 10** QR Algorithm

- $A_0 := A$
- **for** $k = 0, 1, \cdots$ **do**
  - Calculate QR Decomposition $A_k = Q_kR_k$
  - $A_{k+1} := R_kQ_k$
- **end for**

**Algorithm 11** QR Algorithm with shift

- $A_0 := A$
- **for** $k = 0, 1, \cdots$ **do**
  - Choose a shift $\mu_k$ (wilkinson shift)
  - Calculate QR Decomposition $A_k - \mu_kI = Q_kR_k$
  - $A_{k+1} := R_kQ_k + \mu_kI$
- **end for**

**Definition A.1.3.** The minimal polynomial $p$ of a vector $x$ w.r.t. $A$ is the nonzero polynomial of minimal degree such that $p(A)x = 0$.

The degree of the minimal polynomial is called grade of $x$ w.r.t. $A$.

**Corollary A.1.1.** $\mathcal{K}_i(A, x) = \text{span}\{x, Ax, A^2x, \cdots, A^{i-1}x\}$ has dimension $i$ if and only if the grade of $x$ is larger than $i - 1$.

Within Arnoldi process the subspace is orthonormalised and in this orthonormal basis, the matrix operator is represented by an upper Hessenberg matrix $H^{(i)}$ whose eigenvalues yield Ritz approximations to several eigenvalues. Hence, let $x^{(1)}$ be the starting vector and let $X^{(i)} = (x^{(1)}, \cdots, x^{(i)})$ be an
orthonormal basis of \( K_i(A, x^{(1)}) \). Then the subspace in step \( i \) is expanded by calculating \( Ax^{(i)} \) and orthonormalising the result against \( \text{span}\{x^{(1)}, \ldots, x^{(i)}\} \).

It can be shown that this procedure is the same as calculating \( A'x^{(1)} \) and orthonormalising this vector.

Now, the Ritz pairs are calculated from the upper Hessenberg matrix \( X^{(i)H}AX^{(i)} = H^{(i)} \), whose elements are generated during the orthonormalisation process in the Arnoldi algorithm. The Arnoldi process can be written in the form

\[
AX^{(i)} = X^{(i)}H^{(i)} + x^{(i+1)}h_{i+1,i}e_i^H.
\]

If the Arnoldi process breaks down, that is \( h_{i+1,i} = 0 \), we have found an invariant subspace and with \( AX^{(i)} = X^{(i)}H^{(i)} \) the eigenvalues of \( H^{(i)} \) are eigenvalues of \( A \) by a similarity transform. If this is not true we get at least the following error estimate for the Ritz values: \( (\theta, y) (\| y \| = 1) \) is an eigenpair of \( H^{(i)} \) obtained by the Rayleigh Ritz procedure applied to \( A \) and \( X^{(i)} \) then, with \( z = X^{(i)}y \) we have

\[
\| Az - \theta z \| = | h_{i+1,i} | y_i |.
\]

where \( y_i \) denotes the last component of \( y \). Typically, loss of numerical orthogonality of the vectors in \( X^{(i)} \) takes place, requiring reorthogonalisation.

\[
\textbf{Algorithm 12 Arnoldi Algorithm}
\]

\begin{verbatim}
Input \( A, x^{(1)}, (\| x^{(1)} \| = 1), i_{\text{max}} \)
for \( i = 1 \) to \( i_{\text{max}} \) do
    \( x^{(i+1)} = Ax^{(i)} \)
    for \( j = 1 \) to \( i \) do
        Compute \( h_{ji} = x_j^Hx^{(i+1)} \)
        \( x^{(i+1)} = x^{(i+1)} - x^{(j)}h_{ji} \)
    end for
    reorthogonalize
    \( h_{i+1,i} = \| x^{(i+1)} \| \)
    if \( h_{i+1,i} = 0 \) then
        span\{\( x^{(1)}, \ldots, x^{(i)} \}\} is invariant under \( A \)
    end if
    \( x^{(i+1)} = \frac{x^{(i+1)}}{h_{i+1,i}} \)
    \( H^{(i)} = (h_{kj})_{1 \leq k,j \leq i} \)
    \( X^{(i)} = (x^{(1)}, \ldots, x^{(i)}) \)
end for
Output \( H^{(i_{\text{max}})}, X^{(i_{\text{max}})} \)
\end{verbatim}
Remark A.1.2. If $A^H = A$ then also $H^{(i)H} = H^{(i)}$ and so $H^{(i)}$ is tridiagonal and then often called $T^{(i)}$. It simplifies the Arnoldi algorithm, which is then called Lanczos algorithm and significantly reduces the amount of storage, since due to a tree-term recursion, only three vectors are stored at each Lanczos step.

Algorithm 13 Lanczos Algorithm

Input $A, x^{(1)}, (\| x^{(1)} \| = 1), i_{\text{max}}$

for $i = 1$ to $i_{\text{max}}$ do
  $x^{(i+1)} = Ax^{(i)}$
  $\alpha_i = x^{(i)H}x^{(i+1)}$
  $x^{(i+1)} = x^{(i+1)} - \alpha_i x^{(i)}$
  if $i > 1$ then
    $x^{(i+1)} = x^{(i+1)} - \beta_{i-1} x^{(i-1)}$
  end if
  reorthogonalize
  $\beta_i = \| x^{(i+1)} \|$
  if $\beta_i = 0$ then
    $\text{span}\{ x^{(1)}, \cdots, x^{(i)} \} \#$ invariant under $A$
  end if
  $x^{(i+1)} = \frac{x^{(i+1)}}{\beta_i}$
  $T^{(i)} = \text{tridiag}(\alpha_j, \beta_k)^{1 \leq j \leq i, 1 \leq k \leq i}$
  $X^{(i)} = (x^{(1)}, \cdots, x^{(i)})$
end for

Output $T^{(i_{\text{max}})}, X^{(i_{\text{max}})}$

A.2 iterative solvers for linear systems

In this chapter we will describe CG and GMRES, two important iterative solvers for linear systems given by

$$Bz = b$$

(A.1)

where $B \in \mathbb{C}^{n \times n}$, $b$ is a column vector and $z$ is the sought solution. CG and GMRES belong to the class of Krylov subspace methods.

A.2.1 CG for Hermitian positive definite systems

The Conjugate Gradient (CG) algorithm is one of the best known algorithms for solving sparse Hermitian positive definite systems. Let $z^*$ be the exact
solution of (A.1) and the \( z_k \) be the \( k \)-th iterative of some iterative solution technique. The error \( e_k \) and the residual \( r_k \) at step \( k \) are given by

\[
e_k = z^* - z_k
\]

and

\[
r_k = b - Bz_k
\]

respectively. The CG algorithm aims to minimise the \( B \)-norm of the error \( \| e_k \|_B = \sqrt{e_k^H B e_k} \) (note that this is only defined for positive definite \( B \)) over the affine space \( z_0 + \mathcal{K}_k(B, r_0) \) where the \( k \)-th Kyrlov subspace \( \mathcal{K}_k(B, r_0) \) is given by

\[
\mathcal{K}_k(B, r_0) = \text{span}\{r_0, Br_0, \ldots, B^{k-1}r_0\}.
\]

In one sentence the algorithm is an orthogonal projection technique for the error \( z^* - z_k \) onto the Kyrlov subspace \( \mathcal{K}_k(B, r_0) \) and satisfies the Galerkin condition \( r_k \perp \mathcal{K}_k(B, r_0) \). The following algorithm achieves this projection.

**Algorithm 14** CG Algorithm

\[
\begin{align*}
\text{Input} & \quad B, b, z_0, k_{\text{max}}, \varepsilon \\
\text{Output} & \quad z_k \\
\text{for} & \quad k = 1 \text{ to } k_{\text{max}} \text{ do} \\
& \quad \text{if } r_{k-1} < \varepsilon \| b \|_2 \text{ then} \\
& \quad \quad \text{algorithm converged after } k - 1 \text{ iterations.} \\
& \quad \text{end if} \\
& \quad \text{Compute } Bp_{k-1} \\
& \quad \alpha_{k-1} = \frac{\langle r_{k-1}, r_{k-1} \rangle}{\langle p_{k-1}, Bp_{k-1} \rangle} \\
& \quad z_k = z_{k-1} + \alpha_{k-1}p_{k-1} \text{ (update iterate)} \\
& \quad r_k = r_{k-1} - \alpha_{k-1}Bp_{k-1} \text{ (update residual)} \\
& \quad \beta_{k-1} = \frac{\langle r_k, r_k \rangle}{\langle r_{k-1}, r_{k-1} \rangle} \\
& \quad p_k = r_k + \beta_{k-1}p_{k-1} \text{ (update direction vector)} \\
\text{end for} \\
\end{align*}
\]

As result of the algorithm is

\[
\begin{align*}
& r_i^H r_j = 0, \quad p_i^H Bp_j = 0, \text{ for } 0 \leq i, j \leq k - 1, \quad i \neq j.
\end{align*}
\]

that is the residuals \( r_j \) form an orthogonal basis of \( \mathcal{K}_k \) and the search direction \( p_j \) are a conjugate basis. Note that only one matrix-vector multiplication is needed in the algorithm during each iteration, the remaining of the method just consists of inner products.

Since \( z_k \) minimizes \( \| z^* - z_k \|_B \) over \( z_0 + \mathcal{K}_k(B, r_0) \),
\[ \| z^* - z_k \|_B \leq \| z^* - w \|_B \]

holds and with \( w \in z_0 + K_k \) we can write \( w = \sum_{j=0}^{k-1} \gamma_j B^j r_0 + z_0 \) and hence

\[ \| z^* - z_k \|_B = \min_{p \in \Pi_k, p(0)=1} \| p(B)(z^* - z_0) \|_B \]

where \( \Pi_k \) denotes the set of polynomials of degree \( k \).

### A.2.2 GMRES

The method of generalized minimum residuals (or GMRES) was suggested in 1986 by Saad and Schultz. While application of the classical iterative solvers was limited to either diagonally dominant or positive definite matrices, the GMRES method can be used for arbitrary linear systems. The essential ingredient in this general iterative solver is Arnoldi iteration.

The main idea of the GMRES method is to solve a least squares problem at each step of the iteration. More precisely, at step \( n \) we approximate the exact solution \( z^* = B^{-1} b \) by a vector \( z_n \in \mathcal{K}_n(B,b) \) (the \( n \)-th order Krylov subspace) such that residual

\[ \| r_n \|_2 = \| b - Bz_n \|_2 \]

is minimized.

We now describe how to solve this least squares problem. We start with the Krylov matrix:

\[ K_n = [b, Bb, B^2b, \ldots, B^{n-1}b] \in \mathbb{C}^{m \times n}. \]

Thus the column space of \( BK_n \) is \( B\mathcal{K}_n \).

Now the desired \( z_n \in \mathcal{K}_n \) can be written as \( z_n = K_n c \) for some appropriate vector \( c \in \mathbb{C}^n \). Therefore the residual minimization becomes

\[ \| r_n \|_2 = \| b - Bz_n \|_2 = \| BK_n c - b \|_2 \rightarrow \min. \]

The obvious way to find the least squares solution to this problem would be to compute the QR factorization of the matrix \( BK_n \). However, this both unstable and too expensive.

Instead, we look for an orthonormal basis for the Krylov subspace \( \mathcal{K}_n \). With this new basis the approximate solution \( z_n \in \mathcal{K}_n \) can be written as \( z_n = Q_n y \) for some appropriate vector \( y \in \mathbb{C}^n \). The residual minimization is then

\[ \| Bz_n - b \|_2 = \| BQ_n y - b \|_2 \rightarrow \min \quad (A.2) \]

It is now time to recall the principle behind the Arnoldi iteration. That algorithm is based on the partial similarity transform \( BQ_n = Q_{n+1} H_n \),
where \( H_{n+1} \in \mathbb{C}^{n+1 \times n} \) is upper hessenberg matrix. Thus (A.2) turns to 
\[ \| Q_{n+1} H_n y - b \|_2 \rightarrow \min. \]

Next, we take advantage of the fact that multiplication by a unitary matrix
does not change the 2-norm. Thus, we arrive at
\[ \| Q_{n+1}^* Q_{n+1} H_n y - Q_{n+1}^* b \|_2 \rightarrow \min. \]
\[ \iff \| H_n y - Q_{n+1}^* b \|_2 \rightarrow \min. \]

Note that the system matrix \( BQ_n \) in (A.2) is an \( m \times n \) matrix, while the
new matrix \( H_n \) is an \( (n+1) \times n \) matrix which is smaller, and therefore will
permit a more efficient solution. The final simplification we can make is for
the vector \( Q_{n+1}^* b \). In detail, this vector is given by
\[
Q_{n+1}^* b = \begin{pmatrix}
q_1^* b \\
q_2^* b \\
\vdots \\
q_{n+1}^* b 
\end{pmatrix}
\]

Recall that the Krylov subspaces are given by
\[
K_1 = \text{span}\{b\} \\
K_2 = \text{span}\{b, Bb\} \\
\vdots
\]

and that the columns \( q_j \) of \( Q_n \) form an orthonormal basis for \( K_n \). Thus
\( q_1 = \frac{b}{\|b\|} \) and \( q_j^* b = 0 \) for \( j > 1 \). Therefore we actually have \( Q_{n+1}^* b = \| b \| e_1 \)

Combining all of this work we arrive at the final least squares formulation
\[ \| H_n y - b \|_2 \rightarrow \min, \text{ with } z_n = Q_n y. \]

**Algorithm 15 GMRES Algorithm**

Let \( q_1 = \frac{b}{\|b\|} \)

for \( n = 1, 2, 3, \ldots \) do

perform step \( n \) of Arnoldi iteration, i.e., compute a new entries for \( H_n \)
and \( Q_n \).

Find \( y \) that minimizes \( \| H_n y - b \|_2 \| e_1 \|_2 \). (e.g. with QR factorization).

set \( z_n = Q_n y \)

end for
Bibliography


Hiermit versichere ich, die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt zu haben.

Düsseldorf, den. 08.10.2008

Ammar aljammaz