Contents

Introduction

This report gives as an overview of the literature study for the research on computation of thermo-acoustic modes in combustors. Combustors are chambers in which gas is burned. In this process, pressure release causes acoustic effects in the chamber related to the boundaries of the chamber. Especially when impedance occurs at the boundary, this could cause problematic oscillations. To prevent uncontrolled growth of these oscillations, we must know the eigenfrequencies and eigenmodes of these oscillations. These can be found using the Helmholtz equation, which yields an eigenvalue problem. Following this research, we will investigate methods that solve this eigenvalue problem efficiently. This second part of the research will be performed as a practical study within the framework of the Master's thesis. It will be conducted at the Computational Fluid Dynamics department of CERFACS in Toulouse. CERFACS stands for the french translation of European Center for Research and Advanced Training in Scientific Computation. It is "one of the world's leading research institutes working on efficient algorithms for solving large scale scientific problems."

In the first section, an extensive research on the solution of standard eigenvalue problems through subspace methods is performed. Special attention is paid to the methods of Arnoldi and Jacobi-Davidson, because these two methods will be compared in terms of numerical efficiency during the practical study. The second section is dedicated to describing the extension of those methods to the case of generalized and quadratic eigenvalue problems.

In the third section, we consider the solution of nonlinear eigenvalue problems. Since studying the efficiency and possible improvements of the methods that solve such problems is one of the goals of the practical study, this section is to be regarded as the most important part of the literature study. Also, in the practical study we will develop improvements of the existing methods and compare the results. We first describe two methods to solve small nonlinear eigenproblems. These methods are needed to obtain the Ritz pairs for subspace methods like Arnoldi and Jacobi-Davidson, described later in the same section.

Finally, we turn our attention to the application that we will study during the practical study. A brief overview is given of different possibilities for dealing with the problem of coupling between acoustics and heat release in a combustor. We focus on the method using a Helmholtz equation, yielding a nonlinear eigenvalue problem for the oscillation frequency and pressure fluctuation.

In the last section, we will define the problem definition for the rest of the master's thesis project.

1 Standard Eigenvalue Problems

An eigenvector of an operator is a non-null vector whose absolute direction remains unchanged when the operator is applied to that vector. The length of the vector may be changed, even the sign of the direction. The factor with which the vector is multiplied is called the eigenvalue.

The standard eigenvalue problem is how to find pairs of eigenvalues and eigenvectors for a given linear transformation. We can write this in matrix notation as follows:

$$Av = \lambda v$$

where A is the square matrix associated with the transformation, v is the eigenvector and λ is the eigenvalue. Another way to describe the eigenvalues is as the roots of the characteristic polynomial of A: Since $(A - \lambda I)v = 0$ we must have that $\det(A - \lambda I) = 0$. From basic algebra we know that a polynomial has n roots, this shows that any $n \times n$ matrix has n eigenvalues. However, these eigenvalues need not be distinct. In some cases A has less than n distinct eigenvectors. We call such a matrix defective.

Although the mathematical formulation is simple, solving it is hard, especially for large matrices. But since the problem arises from many practical situations studying it is very important. That is why several solution methods have been developed. We will briefly describe three well-known methods, namely the Power method, the Arnoldi method and the Jacobi-Davidson method.

1.1 Power Method

The Power Method is the most basic method to approximate eigenvectors and eigenvalues. The information given here about the power method has largely been taken from [11]. The method creates a sequence of vectors by multiplying by A and scaling. Or mathematically: $v_k = \frac{1}{\alpha_k} A v_{k-1}$. It is easily seen that if the sequence converges, then it must converge to an eigenvector. If we write $v = \lim_{k\to\infty} v_k$ then:

$$Av = A \lim_{k\to\infty} v_k = \lim_{k\to\infty} Av_k = \lim_{k\to\infty} \alpha_k v_{k+1} = \lim_{k\to\infty} \alpha_k \lim_{k\to\infty} v_{k+1} = \lim_{k\to\infty} \alpha_k v.$$

This means that we immediately have found an eigenvalue, namely the limit of α_k. It can be shown that the sequence only converges under the assumption that there is one and only one eigenvalue with the largest modulus. Unfortunately, this can't be checked beforehand. Another draxback of this method is that when the starting vector has no component in the direction of the eigenvector associated with the eigenvalue with the largest modulus then the sequence will not converge to that eigenvector.

Convergence speed is another problem for the power method. To explain this clearly we will first point out a nice property of eigenpairs. Since we know that $Av_k = \lambda_k v_k$, we can combine this for all k in a matrix equation (assuming that A is not defective):

$$[Av_1, \ldots, Av_k] = [\lambda_1 v_1, \ldots, \lambda_k v_k] \;\Rightarrow\; AV = V\Lambda$$

where Λ is the diagonal matrix containing the eigenvalues. This shows we can write $A = V\Lambda V^{-1}$ so $A^k = V\Lambda^k V^{-1}$, which implies that multiplying a vector by A^k will be dominated by the largest eigenvalues. This means that convergence speed depends on how much larger the largest eigenvalue is than the second-largest one. If the relative difference is small, convergence will be slow. It also shows that the power method will only converge to the eigenvalue with the largest modulus.

To make using the Power Method more attractive, a few possible amendments have been developed. We will introduce the Shift-and-Invert technique and Deflation.

1.1.1 Shift-and-Invert

The Shift-and-Invert technique is designed to improve performance of the power method, and to make it possible to find other eigenvalues than the one with the largest modulus. It uses two properties of matrices: If (λ, x) is an eigenpair of A then $(\lambda + \sigma, x)$ is an eigenpair of $(A + \sigma I)$, and λ^{-1} is an eigenvalue of A^{-1}. Adding a constant diagonal is called *shifting* of a matrix, and σ is the *shift*. It is clear that when the shift is close to an eigenvalue λ_k of A then $(\lambda - \sigma)^{-1}$ is the largest eigenvalue of $(A - \sigma I)^{-1}$. So, if we can efficiently use a method on the matrix $(A - \sigma I)^{-1}$ we can target certain eigenvalues. The closer the shift is to λ, the better the convergence. However, if the shift is chosen equal to an eigenvalue, the matrix will be singular. This is no problem, since we have found the desired eigenvalue.

It sounds like a sensible idea to change the shift during the algorithm. This has a drawback however. Because we invert the matrix, we will have to solve a system (rather than calculating the inverse). This is best done with an LU-decomposition. This may seem strange. Since we want the shift close to a desired eigenvalue, our matrix $A - \sigma I$ will be nearly singular and the decomposition ill-conditioned. However, since the error will only be large in the direction of the desired eigenvector, this has no effect on the convergence speed. Remember that a multiple of an eigenvector is again an eigenvector.

Every time the shift is changed, we will have to re-calculate this decomposition. Note that decomposing a matrix is more costly than solving a triangular system, so there is a trade-off in convergence speed between the amount of work needed per iteration and the number of iterations until convergence. This also depends on the size of A. The larger the matrix, the more work is needed to decompose it. In case A is very large it may be better to use an iterative method to solve the system. But then, changing the shift so that it is closer to an eigenvalue will increase the problems that arise because of the singularity of the matrix $A - \lambda I$. The most important problem is that iterative methods are generally based on multiplying vectors by A, or in this case the shifted version $A - \sigma I$. The closer σ is to λ the *slower* these iterative methods will converge, even though the two largest eigenvalues of $(A - \sigma I)^{-1}$ are further apart the closer σ is to λ. This is a crucial weakness of the shift-and-invert method.

1.1.2 Deflation

A second technique to change the matrix such that another eigenvalue has the largest modulus is the so-called *deflation* technique. It is similar to the shift-and-invert technique in that it adds a matrix to A to change the eigenvalues, but it does not invert. The idea is to add a matrix that changes only one eigenvalue. To use it, only the right eigenvector corresponding to the eigenvalue with largest modulus is needed. We shift the matrix A by substracting $\sigma u_1 v^H$ where u_1 is the known eigenvector and v^H is the hermitian transpose of a vector v such that $v^H u_1 = 1$. σ is the desired shift: the eigenvalues of $A - \sigma u_1 v^H$ will be $\{\lambda_1 - \sigma, \lambda_2, \ldots, \lambda_n\}$. In [11] it has been shown empirically that, assuming the eigenvectors are normalized when they are found, choosing $v = u_1$ speeds up convergence in comparison to choosing a random vector with the property that $v^H u_1 = 1$. When already m eigenpairs are known, we can create an $n \times m$ matrix Q containing the eigenvectors as columns, and an $m \times m$ diagonal matrix Σ with the desired shifts as non-zero elements. It can be shown that the eigenvalues of the shifted matrix $A - Q\Sigma Q^H$ are $\{\lambda_1 - \sigma_2, \ldots, \lambda_m - \sigma_m, \lambda_{m+1}, \ldots, \lambda_n\}$. We can use this technique to find the eigenvalues of A, starting with the one with largest modulus and working our way down.

1.2 Search spaces and Ritz Values

Although the techniques described above make the Power method more powerful, it still has a weakness. The Power method only uses the last approximation to compute a new one. This means that all information from previous approximation is not used. An obvious improvement would be to work with subspaces. Important methods that use subspaces are Arnoldi and Jacobi-Davidson. As a supporting technique, we introduce the concept of Ritz values: θ_k is Ritz value of A with respect to the subspace $\mathcal{V}_k$ with Ritz vector u_k if

$$u_k \in \mathcal{V}_k, u_k \neq 0,\ Au_k - \theta_k u_k \perp \mathcal{V}_k.$$

This means that if we solve the projected eigenvalue problem on the subspace $\mathcal{V}_k$, we find Ritz values of A. A Ritz pair (u_k, θ_k) is a solution of the equation

$$W_k^* A V_k u_k = \theta_k W_k^* V_k u_k$$

which is found by writing $x = V_k u_k$ and multiplying the equation by an appropriate matrix W_k^*. Usually, we choose $W_k = V_k$.

V_k is a matrix of column vectors spanning $\mathcal{V}_k$. Note that we can choose V_k to consist of the orthonormal basis of $\mathcal{V}_k$, in which case $V_k^* V_k = I$ and the problem above reduces to an $k \times k$ eigenvalue problem: $(V_k^* A V_k - \theta I)u_k = 0$.

1.3 Basic Arnoldi

A more sophisticated algorithm than the power method is the Arnoldi method. This method originally was developed last century by Arnoldi in [1] to transform dense matrices into Hessenberg form. Arnoldi himself already noted that this method could be used to approximate certain eigenvalues, even without finishing the transformation algorithm. Later on, the method was used to find eigenvalues of large sparse matrices.

Arnoldi's method is a Krylov subspace method. Krylov subspaces are based on the simple power method. They are formed as followed: first a starting vector is chosen, say v_0. Then the Krylov subspace is defined as:

$$\mathcal{K}_k(A, v) \equiv \text{span}\{v, Av, A^2v, \ldots, A^{k-1}v\}.$$

Notice that $\mathcal{K}_k$ is k-dimensional, while $A \in \mathbb{R}^{n\times n}$. Of course, k is always chosen (much) smaller than n.

Arnoldi's method creates k orthonormal vectors that form a basis for $\mathcal{K}_k$. The vectors are combined in to an $n \times m$-matrix V_k, while an $k \times k$-Hessenberg matrix H_k is formed by $h_{ij} = (Av_j, v_i)$. The exterior eigenvalues of H_k can be used as approximations of eigenvalues of A. Usually, this will be a small fraction of the k eigenvalues of H_k. Note that by construction, $H_k = V_k^* A V_k$.

The basic algorithm is as follows:

Algorithm 1 Arnoldi

```
1:  Start: Choose an initial vector $v_1$ of length one,
2:  for $j = 1, 2, \ldots, m$ do
3:      $h_{ij} = (Av_j, v_i),\ i = 1, 2, \ldots, j$, {1}
4:      $w_j = Av_j - \sum_{i=1}^{j} h_{ij}v_i$, {2}
5:      $h_{j+1,j} = ||w_j||_2$,
6:      if $h_{j+1,j} = 0$ then
7:          stop
8:      end if
9:      $v_{j+1} = w_j / h_{j+1,j}$.
10: end for
```

At line 3, we have to do a matrix-vector multiplication. This is a costly operation. We can store the result to use it again at line 4, where we perform Gram-Schmidt orthogonalization. In practice, this orthogonalization will be modified Gram-Schmidt. Keep in mind that this algorithm is only a very simple version, that only shows the idea of using a Krylov subspace as a search space for eigenvectors. For practical implementation, a lot of improvements can be made. The method of orthogonalization could be improved, and shift-and-invert and deflation techniques could be combined with arnoldi.

1.3.1 Lanczos' Method

A method quite similar to Arnoldi's is the Lanczos Method [5]. The difference is that Lanczos assumed A to be Hermitian, in which case $H_k = V_k^* A V_k = V_k^* A^* V_k = H_k^*$, so H_k is also Hermitian. Since H_k is known to be a Hessenberg matrix, of which the elements h_{ij} are zero if $j \geq i + 2$, it follows that H_k is in this case tri-diagonal. Lanczos' method uses this fact to speed up calculations. Although Lanczos' method is an important and interesting method, is is outside the scope of this study and will not be further investigated

1.3.2 Restart

When working with a Krylov-subspace algorithm, we build a subspace from approximations of eigenvectors. When convergence is slow, for instance due to the starting vector, the subspace will continue to grow. This gives two problems. The first is that a growing subspace means larger matrices to work with, and this increases the cost in terms of computation work and time quadratically. The second problem is that we need a lot of memory to store all the information we have about the subspace. To solve these problems, a technique called *restart* has been developed. This is a general name for techniques that reduce the size of the Krylov subspace when a certain size is reached. Some methods restart with a one dimensional subspace spanned by the latest approximation of the eigenvector. Other methods use several approximations as a basis of the starting subspace. Especially when working with the Jacobi-Davidson method, that will be introduced in the next section, it is common to restart with a subspace with dimension larger than 1. For Arnoldi's method, extensive research has been done to improve performance with clever restarts. Many of the results are implemented in the well-known Computational package ARPACK [6], that is used in e.g. MATLAB to calculate eigenvalues.

1.4 Jacobi-Davidson

Similar to the Arnoldi method, Jacobi-Davidson is a subspace method. It was first published in [14]. However, it does not use a Krylov subspace as search space for the eigenvector. In this method, we also construct a matrix V_k and H_k, and use eigenvalues of H_k as approximations of the eigenvalues of A. The main difference is the way in which the matrix V_k is constructed. Instead of using the power method to create a Krylov subspace, we combine ideas of Jacobi and Davidson to look at the orthogonal projection of A onto the complement of our current approximation u_k to create the search space. Before we go deeper into the Jacobi-Davidson method, we will introduce the Jacobi and the Davidson method.

1.4.1 Jacobi

Jacobi published in [4] a method to find eigenvalues of a diagonally dominant matrix A of which the largest diagonal element is $a_{1,1} = \alpha$. The idea is to write the eigenvalue problem as:

$$A \begin{bmatrix} 1 \\ z \end{bmatrix} = \begin{bmatrix} \alpha & c^T \\ b & F \end{bmatrix} \begin{bmatrix} 1 \\ z \end{bmatrix} = \lambda \begin{bmatrix} 1 \\ z \end{bmatrix}.$$

This can be written as a system of two equations, writing θ as the approximation of λ:

$$\begin{aligned} \alpha + c^T z &= \theta \\ (F - \theta I)z &= -b \end{aligned}$$

If we start with a vector z we can compute θ from the first equation, and insert this value in the second equation and solve for z, thus iteratively approximating λ by θ. To use Jacobi for an arbitrary matrix A, one needs to diagonalize the matrix first (or at least do a few steps in order to make it a diagonally dominant matrix) and exchange some rows and columns to get the largest diagonal-value on the right place.

1.4.2 Davidson

Davidson's method [2] creates a subspace that is build from subsequent approximations of the desired eigenvector. Suppose that we have a subspace $\mathcal{K}$ of dimension k, with basis $v_1, \dots, v_k$. We can compute the Ritz-value θ_k and the Ritz vector u_k of the matrix A over this subspace. The residual is $r_k = Au_k - \theta_k u_k$. The method of expanding the search space is first to compute $t = (D_A - \theta_k I)^{-1} r_k$. Here, D_A is the diagonal of A, which is chosen because the cost of inverting $(D_A - \theta_k I)$ is drastically lower than for $(A - \theta_k I)$. Notice that in this way, we are actually approximating a shift-and-invert step. But because we use a diagonal approximation of $(A - \theta_k I)$, we don't have to worry about decomposing, avoinding shift-changes or iteratively solving for t. The vector t is made orthogonal to $v_1, \dots, v_k$, and the resulting vector will be v_{k+1}, expanding $\mathcal{K}$. u_k will approximate an eigenvector of A, and θ_k again approximates the corresponding eigenvalue.

1.4.3 Jacobi-Davidson

The Jacobi-Davidson method finds inspiration in ideas from both methods described above. Starting with an initial guess, we search for a correction for the approximate eigenvector in the directions orthogonal to the current approximation, that is, in the subspace $u_k^\perp$. We will do so by first finding an approximation for the eigenvalue using the current approximation of the eigenvector. Then we use the approximated eigenvalue to find an approximation of the eigenvector. Clearly, this idea is similar to Jacobi's.

The approximation θ_k for the eigenvalue λ is found as follows:

$$Ax = \lambda x \Rightarrow x^* Ax = x^* \lambda x$$

Inserting u_k as approximation for x we will define $\theta_k = u_k^* A u_k$, where we assume that u_k has been normalized.

To approximate the eigenvector, we use the idea from Davidson to look in the subspace $u_k^\perp$. We want to find a correction $v \in u_k^\perp$ such that:

$$\begin{aligned} A(u_k + v) &= \lambda(u_k + v) \\ \Rightarrow\ (A - \lambda I)v &= -(A - \lambda I)u_k = -r_k + (\lambda - \theta_k)u_k \qquad (1) \end{aligned}$$

where $r_k = (A - \theta_k I)u_k$ is the residual from the latest approximation. We project this equation on the subspace $u_k^\perp$ by multiplying on the left side by $I - u_k u_k^*$. We use the following observations:

$$\begin{array}{rcl} (I - u_k u_k^*)v &=& v, \\ (I - u_k u_k^*)r_k &=& (I - u_k u_k^*)(A - \theta_k I)u_k \\ &=& r_k - u_k(u_k^* A u_k - \theta_k) \\ &=& r_k, \\ (I - u_k u_k^*)u_k &=& 0 \end{array} .$$

If we now multiply (1) with $I - u_k u_k^*$ and replace the unknown λ by the known approximation θ_k, we find the so called *Jacobi-Davidson correction equation:*:

$$(I - u_k u_k^*)(A - \theta_k I)(I - u_k u_k^*)v = -r_k \qquad (2)$$

Since the $(A - \theta_k I)$ is transformed to be in $u_k^\perp$ the rank is less than n and the equation is in fact ill-posed. But we are only interested in the part of the solution that is in the same direction as v, so we will use this equation to iteratively approximate the correction v by $\hat{v}$, and the next approximation for the eigenvector will be $u_k + \hat{v}$. A popular method to use when solving iteratively is GMRES [12].

It is clear that the correction equation is similar to a shift-and-invert step. Remember that a shift-and-invert step is to find an update by multiplying the current approximation by the shifted and inverted matrix: $(A - \theta_k I)^{-1} v_k = v_{k+1}$. The difference between Jacobi-Davidson and Shift-and-invert is that Jacobi-Davidson restricts itself to a particular search direction, avoiding the problem of singularity when the shift is close to an eigenvalue.

1.4.4 Search spaces and Harmonic Ritz values

An obvious adaption of the Jacobi-Davidson method is to use the correction v not to correct the current approximation u_k directly, but to store the k-th correction as v_k. We can then build a subspace $\mathcal{V}_k = \text{span}\{v_1, \dots, v_k\}$ of dimension k and use it to compute Ritz pairs of A. In this way, we obtain approximations of several eigenpairs at once. We can choose to focus on one value, or we can try to let more than one value converge. A problem with this technique is that it will only converge to exterior eigenvalues. However, this can easily be solved by shifting the matrix A. To keep convergence speed high, we can use so-called *harmonic Ritz values*. Harmonic Ritz values were introduced in [10]. μ_k is a harmonic Ritz value of A with respect to some linear subspace $\mathcal{W}_k$ if μ_k^{-1} is a Ritz value of A^{-1} with respect to $\mathcal{W}_k$. If we choose $\mathcal{W}_k = \text{span}\{Av_1, \dots, Av_k\}$, $W_k = AV_k$, $x = AV_k y$ and $\mu = \lambda^{-1}$ then we can write, starting from the original eigenvalue problem:

$$\begin{array}{rrcl} & Ax & = & \lambda x \\ \Rightarrow & \mu x & = & A^{-1}x \\ \Rightarrow & \mu AV_k y & = & V_k y \\ \Rightarrow & \mu W_k^* W_k y & = & W_k^* V_k y \end{array}$$

We now have a generalized eigenvalue problem, which we will treat later in this paper. Solving this problem for μ will give us an approximation of the smallest eigenvalues of the original problem. By combining harmonic Ritz values with a shift of the matrix (see section 1.1.1) we can efficiently obtain interior values.

1.4.5 The Jacobi-Davidson Algorithm

The basic algorithm for calculation of a single eigenvalue of the standard eigenproblem using Jacobi-Davidson is provided by Gerard L.G. Sleijpen et al in [15]. The algorithm is:

Algorithm 2 Jacobi-Davidson Method for λ_{max} of A

```
1: Start with $t = v_0$, starting guess
2: for i=1,...,k-1 do
3:     $t = t - (t^* v_i) v_i$
4: end for
5: $v_k = t/||t||_2$
6: $v_k^A = A v_k$
7: for i=1,...,m-1 do
8:     $M_{i,k} = v_i^* v_k^A$
9:     $M_{k,i} = v_k^* v_i^A$
10: end for
11: $M_{k,k} = v_k^* v_k^A$
12: Compute the largest eigenpair of the eigenproblem $Ms = \theta s$ of the $k \times k$ matrix $M$,
    ($||s||_2 = 1$)
13: $u = Vs$
14: $u^A = V^A s$
15: $r = u^A - \theta u$
16: if $||r||_2 \leq \epsilon$ then
17:     $\tilde{\lambda} = \theta, \tilde{x} = u$
18:     STOP
19: end if
20: Solve $t \perp u$ (approximately) from $(I - uu^*)(A - \theta I)(I - uu^*)t = -r$
```

There are three important points in the algorithm that need some explaining. First, the latest expansion to the basis of the search space is orthogonalised using Gram-Schmidt. The basis vectors v_k together form the matrix V. Then, at lines 7-11, we construct the matrix V^*AV, which we use to calculate the Ritz values of A in the next step of the algorithm. At line 20 we solve the correction equation, the fundamental equation of the Jacobi-Davidson theory. Also note (again) that the correction equation is in fact a transformed shift-and-invert step with shift θ. The shift was obtained as a Ritz value at line 11. We can actually target any eigenvalue by choosing another Ritz value. Of course, this means that we need a good method to find the Ritz values. A commonly used method is the QR algorithm for eigenvalues, see [3]. We can solve for the eigenvalues directly in this case because it is typically a small problem (size k).

In the same paper, Gerard L.G. Sleijpen et al proposed a more advanced algorithm based on harmonic Ritz values en vectors, inluding restart and deflation techniques. The algorithm is given on the next page.

Algorithm 3 Jacobi-Davidson Method for k_{max} eigenvalues of A close to τ

```
Start with $t = v_0$, $k = 0$, $m = 0$, $Q = [\,]$, $R = [\,]$.
while $k < kmax$ do
  for $i = 1, \ldots, m$ do
    $t = t - (v_i^* t) v_i$
  end for
  $m = m + 1, v_m = t/||t||_2$, $v_m^A = Av_m - \tau v_m$, $w = v_m^A$
  for $i = 1, \ldots, k$ do
    $w = w - (q_i^* w) q_i$
  end for
  for $i = 1, \ldots, m - 1$ do
    $M_{i,m}^A = w_i^* w$, $w = w - M_{i,m}^A w_i$
  end for
  $M_{m,m}^A = ||w||_2$, $w_m = w/M_{m,m}^A$
  for $i = 1, \ldots, m - 1$ do
    $M_{i,m} = w_i^* v_m$, $M_{m,i} = w_m^* v_i$
  end for
  $M_{m,m} = w_m^* v_m$
  Make a $QZ$ decomposition $M^A S^R = S^L T^A$, $MS^R = S^L T$, $S^R$, $S^L$ unitary and
  $T^A$, $T$ upper triangular, such that: $|T_{i,i}^A / T_{i,i}| \leq |T_{i+1,i+1}^A / T_{i+1,i+1}|$
  $u = Vs_1^R, u^A = V^A s_1^R, \vartheta = \bar{T}_{1,1} \cdot T_{1,1}^A,$
  $r = u^A - \vartheta u, \tilde{a} = Q^* r, \tilde{r} = r - Q\tilde{a}$
  while $||\tilde{r}||_2 \leq \epsilon$ do
    $R = \begin{pmatrix} R & \tilde{a} \\ 0 & \vartheta + \tau \end{pmatrix}$, $Q = [Q, u],\ k = k + 1$
    if $k = k_{max}$ then
      STOP
    end if
    $m = m - 1$
    for $i = 1, \ldots, m$ do
      $v_i = Vs_{i+1}^R$, $v_i^A = V^A s_{i+1}^R$, $w_i = Ws_{i+1}^L$, $s_i^R = s_i^L = e_i$
    end for
    $M^A,\ M$ are the lower $m \times m$-blocks of $T^A$, $T$ respectively
    $u = v_1, u^A = v_1^A, \vartheta = \bar{M}_{1,1} \cdot M_{1,1}^A$
    $r = u^A - \vartheta u, \tilde{a} = Q^* r, \tilde{r} = r - Q\tilde{a}$
  end while
  if $m \geq m_{max}$ then
    for $i = 2, \ldots, m_{min}$ do
      $v_i = Vs_i^R$, $v_i^A = v^A s_i^R$, $w_i W s_i^L$
    end for
    $M^A,\ M$ are the leading $m_{min} \times m_{min}$-blocks of $T^A$, $T$ respectively
    $v_1 = u,\ v_1^A = u^A,\ w_1 = Ws_1^L, m = m_{min}$
  end if
  $\theta = \vartheta + \tau, \widetilde{Q} = [Q, u]$
  Solve $t \perp Q$ (approximately) from $(I - \widetilde{Q}\widetilde{Q}^*)(A - \theta_k I)(I - \widetilde{Q}\widetilde{Q}^*)t = -\tilde{r}$
end while
```

We'll briefly explain some parts. The first part is used to orthogonalize the latest correction t against the current searchspace, and to expand the matrices and vectors accordingly. Then we use a QZ decomposition for both M^A and M using the same matrices S^R and S^L: $M^A S^R = S^L T^A$ and $MS^R = S^L T$. See the next section for more information about the QZ algorithm

After this we compute new approximations for the eigenpair and the residual at line 21, and if we find that norm of the residual has fallen below a given threshold, we are satisfied with the approximation $\vartheta + \tau$ for the eigenvalue. We store this value in the matrix R, together with the vector $\tilde{a}$ such that R is an upper triangle matrix. Another matrix Q is formed such that $AQ = QR$.

From line 23 to line 39 the algorithm checks whether a restart is needed, and if a restart is necessary the variables are changed appropriately. This is the case if either the residual is below a certain threshold ϵ, meaning that we are satisfied with the current approximation, or when the dimension of the search space gets too large. In the first case we restart by removing the converged eigenvector from the basis, so that the dimension of the search space decreases by one. We also need to make appropriate choices for the other variables, see lines 27-32 . If we don't reach the threshold before $m \leq m_{max}$ we restart with $m = m_{min}$, where, m_{min} is the pre-determined minimal number of vectors that forms the new basis. Also here, we need to restrict the other variables, see lines 35-39 . If neither condition is met, then we simply continue by updating θ, $\widetilde{Q}$ and t.

1.5 The QZ decomposition

In the algorithm for Jacobi-Davidson, we used the so called QZ decomposition. This decomposition is a more general form of the *Schur* decomposition. In fact, it is also referred to as the generalized Schur decomposition. The Schur decomposition aims to write a square matrix $A = Q^*UQ$ where Q is a unitary matrix containing the orthogonalized eigenvectors of A and U upper triangular with the eigenvalues of A on the main diagonal. The QZ decomposition decomposes two matrices using the same unitary matrices for both matrices. We will give this statement as a theorem:

Theorem 1.1 *If A and B are in $\mathbb{C}^{n \times n}$, then there exist unitary Q and Z such that $Q^*AZ = T$ and $Q^*BZ = S$ are upper triangular. If for some k, t_{kk} and s_{kk} are both zero, then $\lambda(A, B) = \mathbb{C}$. Otherwise $\lambda(A, B) = t_{ii}/s_{ii} : s_{ii} \neq 0$.*

$\lambda(A, B)$ is the spectrum (subset of $\mathbb{C}$ containing the eigenval ues) of the generalized eigenproblem $Ax = \lambda Bx$ described in the next section. This theorem and it's proof can be found as theorem 7.7.1 in [3], as well as more information about these and other decompositions.

2 Generalized and Quadratic Eigenvalue Problems

The standard eigenvalue problem is to find a vector and a value for a certain transformation such that the transformed vector is the same as the value multiplied with the vector. An obvious generalization of this problem is to transform the right hand side as well, so for a given pair of matrices A, B (representing transformations) we want to find a vector x and value λ such that:

$$Ax = \lambda Bx.$$

If B is I then we get our original eigenvalue problem back. This more general problem is called the *generalized eigenvalue problem*. It is usually preferred to define $\lambda = \alpha/\beta$ and writing:

$$(\beta A - \alpha B)x = 0,$$

since these numbers are even valid when β is zero, while λ will go to infinity in this case. We will call (α, β) an eigenvalue of the problem $(\beta A - \alpha B)x = 0$.

Generalized eigenproblems arise frequently from applications. If we expand the generalized eigenproblem by a quadratic term, we can deal with even more practical problems. This gives us the *quadratic eigenvalue problem*:

$$\lambda^2 Cx + \lambda Bx + Ax = 0$$

We will discuss both types of problems, restricting ourselves to what we need to understand how to apply search space methods to these problems. The theory presented in this section is for a large part taken from [11].

2.1 Generalized Eigenproblems

When dealing with Generalized Eigenproblems, a few problems may occur. For instance, there are infinitely many eigenvalues (α, β), because we can simply multiply with a constant to find another eigenvalue of the same eigenvector. There are several ways to deal with this: we could just define β to be 1, which could be established for any eigenvalue with $\beta \neq 0$ by scaling. Even though the case that $\beta = 0$ is rare, it is not needed to discard the option. A better solution is to scale the found eigenvalues such that $|\alpha|^2 + |\beta|^2 = 1$.

Another problem is that the matrix pair (A, B) may be singular, that is det$(\beta A - \alpha B)$ might be zero for all (α, β). In that case, any pair of numbers is an eigenvalue. This is not very interesting of course. It is important to realize that it is well possible for a matrix pair to be regular (non-singular) even when one or both of the matrices involved are singular. In fact, if either A or B is regular, we can write the Generalized eigenproblem as a standard eigenproblem by multiplying with the inverted regular matrix. In case A is singular, we also need to multiply by $1/\lambda$ to get a standard eigenvalue problem.

2.1.1 Properties of the Generalized Eigenvalue Problem

Before we discuss methods of solving generalized eigenproblems, we will look at two properties of these problems. We will see how certain transformations influence the eigenvectors and eigenvalues of the problem. When we multiply both A and B from the left with the same non-singular matrix Y, the right eigenvectors are preserved, while the left eigenvectors are multiplied by Y^{-*}. The first statement is trivial. The second statement follows from:

$$\begin{array}{rrcl} & v^*(\beta A - \alpha B) & = & 0 \\ \Leftrightarrow & v^* Y^{-1} Y(\beta A - \alpha B) & = & 0 \\ \Leftrightarrow & (Y^{-*}v)^*(\beta Y A - \alpha Y B) & = & 0. \end{array}$$

Here, we have assumed that v is a left eigenvector of the original problem, and 0 denotes the zero row vector of appropriate length. In the same way, we can show that when we multiply both matrices from the right with a non-singular matrix X the left eigenvectors are preserved, while the right eigenvectors are multiplied by X^{-1}. In both cases, the eigenvalues remain the same. We say that for non-singular X and Y the pair (YAX, YBX) is equivalent to (A, B).

The second interesting property is found as a theorem in [11]. The theorem states:

Theorem 2.1 *Let (A, B) be any matrix pair and consider the transformed matrix pair (A_1, B_1) defined by:*

$$A_1 = \tau_1 A - \sigma_1 B,\ B_1 = \tau_2 B - \sigma_2 A$$

for any four scalars $\tau_1, \tau_2, \sigma_1, \sigma_2$ such that the 2×2 matrix

$$\Omega = \begin{pmatrix} \tau_2 & \sigma_1 \\ \sigma_2 & \tau_1 \end{pmatrix}$$

is non-singular. Then the pair (A_1, B_1) has the same eigenvectors as the pair (A, B). An associated eigenvalue $< \alpha^{(1)}, \beta^{(1)} >$ of the transformed matrix pair is related to the eigenvalue $< \alpha, \beta >$ of the original pair by:

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \Omega \begin{pmatrix} \alpha^{(1)} \\ \beta^{(1)} \end{pmatrix}.$$

2.1.2 Projection Methods

Previously, we discussed projection (or search space) methods like Arnoldi and Jacobi-Davidson. We would like to find a way to solve generalized eigenproblems using these methods. In general, projection methods search for an eigenvalue (α, β) with an eigenvector u in a subspace $\mathcal{K}$ such that:

$$(\beta A - \alpha B)u \perp \mathcal{L}$$

for given subspaces $\mathcal{K}$ and $\mathcal{L}$. If we know two bases $V = \{v_1, \ldots, v_m\}$ and $W = \{w_1, \ldots, w_m\}$ of $\mathcal{K}$ and $\mathcal{L}$ respectively, we can rewrite the search space problem as an ordinary generalized eigenproblem of dimension m:

$$(\beta W^H A V - \alpha W^H B V)y = 0$$

were we have written u by Vy. The way in which the subspaces $\mathcal{K}$ and $\mathcal{L}$ are formed depends on the choice of method. Unfortunately, for the Arnoldi method this is not possible, because, as we will show, we cannot construct an appropriate Krylov subspace, so we have no $\mathcal{K}$. To make this more clear, remember that Arnoldi uses a Krylov subspace as a search space. To construct the Krylov subspace, the current approximation is multiplied by A and orthogonalized. In the case of a generalized eigenproblem, multiplying by A would not help because we also have a matrix B. The appropriate Krylov subspace would be $(v, B^{-1}Av, \ldots, (B^{-1}A)^k v)$, which is the Krylov subspace associated with the matrix $B^{-1}A$. In other words, to use Arnoldi we will have to write the problem as a standard eigenvalue problem $B^{-1}Ax = \lambda x$, except when we can find X and Y such that $YBX = I$ (or $YAX = I$), in which case we write $YAX = \lambda x$ where the eigenvectors are transformed as described before. With Jacobi-Davidson, we can easily extend the method to work for generalized or even quadratic problems, as will be shown in the following section. Finally, an outline for an algorithm will be given in the section about nonlinear eigenproblems. This algorith can be easily adapted to fit generalized or polynomial eigenproblems.

2.2 Quadratic Eigenvalue Problems

Quadratic eigenvalue problems are of the form $\lambda^2 Cx + \lambda Bx + Ax = 0$. These problems often arise from practical applications, like a spring system. For Arnoldi, it is impossible to solve these problems directly. Usually, one will rewrite the problem to generalized form (linearization). We will present the most common way of linearization, and then turn our attention to directly solving a polynomial problem with Jacobi-Davidson.

2.2.1 Rewriting into a Generalized Eigenvalue Problem

When dealing with a quadratic eigenproblem it is always possible to rewrite the problem to a generalized eigenproblem. We can rewrite the problem as:

$$\lambda \begin{pmatrix} B & C \\ I & 0 \end{pmatrix} \begin{pmatrix} x \\ \lambda x \end{pmatrix} = \begin{pmatrix} -A & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} x \\ \lambda x \end{pmatrix}.$$

Notice that the Identity Matrix can be replaced by any matrix. This will affect performance of the method that is used to solve the generalized eigenproblem. The most popular alternative is to use C instead of I. This will usually give better performance, especially when all matrices involveld are Hermitian (which happens often when the matrices arise from numerically discretized equations), since then the blockmatrices will also be Hermitian. If C is a symmetric positive definite matrix (which is the case for certain classes of physical problems) it may be better to rewrite the quadratic problem as follows:

$$\lambda \begin{pmatrix} C & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} \lambda x \\ x \end{pmatrix} = \begin{pmatrix} -B & -A \\ I & 0 \end{pmatrix} \begin{pmatrix} \lambda x \\ x \end{pmatrix} = 0.$$

This gives us a symmetric positive definite matrix on the left-hand side which can be inverted cheaply or, by using the right pre- and postmultiplications, be transformed to I. Notice that the size of the problem has increased to $2n$. For general polynomial eigenproblems of degree k the number of eigenvalues is kn.

2.2.2 Direct Solving of the Quadratic Eigenproblem

In one of G.L.G. Sleijpens articles [13] he indicates how to solve a quadratic eigenproblem (or more general: a polynomial eigenproblem) with a Jacobi-Davidson method. The general polynomial eigenproblem is to find a non-trivial eigenvector x and it's associated eigenvalue $\lambda \in C$ such that:

$$A_0 x + \lambda A_1 x + \cdots + \lambda^n A_n x = 0 \tag{3}$$

For ease of notation, we write this in terms of a matrix-valued polynomial:

$$\Psi(\lambda)x = 0 \text{ where } \Psi(\vartheta) = A_0 + \vartheta A_1 + \cdots + \vartheta^n A_n$$

We proceed much like we did for the standard eigenproblem. We suppose we have a m-dimensional search subspace $\mathcal{V}_m$ and a m-dimensional projection subspace $\mathcal{W}_m$. We can then compute an approximation u of x with associated approximation ϑ of λ, by solving the following projected problem:

$$u \in \mathcal{V}_m, \vartheta \in \mathbb{C} \text{ such that } \Psi(\vartheta)u \perp \mathcal{W}_m. \tag{4}$$

The residual r is defined by $r \equiv \Psi(\theta)u$, and for some arbitrary $\tilde{u}$ we correct the approximation u by z_1, where z_1 is an approximate solution of the correction equation:

$$z \perp u \text{ and } \left(I - \frac{\tilde{w}w^*}{w^*\tilde{w}}\right) \Psi(\vartheta) \left(I - \frac{u\tilde{u}^*}{\tilde{w}^*u}\right) z = -r$$

for relevant choices of w and $\tilde{w}$. The speed of convergence depends heavily on the choice of these vectors. In their article, G.L.G. Sleijpen et al have suggested a couple of possibilities. We will present their two most straightforward choices: $\tilde{w} = w = \tilde{u} = u$ giving linear convergence, and $w = \tilde{u} = u$ and $\tilde{w} = T'(\vartheta)$ giving quadratic convergence [13].

3 Nonlinear Eigenvalue Problems

The most difficult challenge in the context of eigenvalue problems is solving the most general form: the nonlinear eigenvalue problem. We write the problem as:

$$T(\lambda)x = 0, \; T : \mathbb{R} \to \mathbb{R}^{n\times n}, \tag{5}$$

where T is a matrix function of λ. This is obviously a generalization of the eigenproblems we have discussed so far. For example, if $T(\lambda) := \lambda^2 C + \lambda B + A$ then (5) reduces to a quadratic eigenproblem. Various physical problems are known to reduce to nonlinear eigenvalue problems, for example in acoustic modelling.

3.1 Numerical Methods for Nonlinear Eigenproblems

We will investigate how the two methods we have been studying so far, Arnoldi and Jacobi-Davidson, can be extended to the nonlinear case. These methods were originally designed for dealing with sparse matrices. For dense nonlinear eigenproblems, several other methods have been developed, some of which are discussed by Mehrmann and Voss in [16] and in [7]. The same articles research the iterative projection methods we are interested in: an Arnoldi-type method for nonlinear eigenproblems, as well as an extension of the Jacobi-Davidson method to the nonlinear case. A third projection is also proposed, namely the Rational Krylov method. We will give a summary of this research, with the exclusion of the Rational Krylov method as this falls outside the scope of our own research.

3.2 Newton Type Methods for Small Dense Problems

Before we turn our attention to subspace methods, we will describe some of the methods presented in [16] and in [7]. We will need knowledge of these methods since we will need to solve a small dense problem to obtain Ritz values for the large sparse problems we solve iteratively.

3.2.1 Inverse Iteration

It is noted in [7] for the linear case where $T(\lambda) = A - \lambda I$ that the so-called inverse iteration method can be found by applying Newton's method to the nonlinear system:

$$\begin{bmatrix} T(\lambda)x \\ v^* x - 1 \end{bmatrix} = 0.$$

If we let $T(\lambda)$ be nonlinear and apply one step of Newton's method to the same system we find:

$$\begin{bmatrix} T(\lambda_k) & T'(\lambda_k)x_k \\ v^* & 0 \end{bmatrix} \begin{bmatrix} x_{k+1} - x_k \\ \lambda_{k+1} - \lambda_k \end{bmatrix} = \begin{bmatrix} T(\lambda_k)x_k \\ v^* x_k - 1 \end{bmatrix}.$$

We can rewrite this in terms of x_{k+1}:

$$\begin{aligned} x_{k+1} &= -(\lambda_{k+1} - \lambda_k)T(\lambda_k)^{-1}T'(\lambda_k)x_k & (6) \\ v^* x_{k+1} &= 1. & (7) \end{aligned}$$

The first expression gives us a direction u_{k+1} for the new approximation of the eigenvector:

$$u_{k+1} := T(\lambda_k)^{-1}T'(\lambda_k)x_k.$$

Multiplying the first equation with v^* and combining with the second gives $1 = -(\lambda_{k+1} - \lambda_k)v^* u_{k+1}$ or:

$$\lambda_{k+1} = \lambda_k - \frac{1}{v^* u_{k+1}}.$$

So we obtain an approximation for both the eigenvalue and (after normalization) the eigenvector.

3.2.2 Residual Inverse Iteration

A drawback of the inverse iteration is that in every iteration we need a LU-decomposition of $T(\lambda)$ in order to find an update direction u_{k+1}. This can be solved by using $T(\sigma)$ instead of $T(\lambda)$, where σ is a fixed shift close to the desired eigenvalue λ. However, Mehrmann and Voss argue that simply substituting this in in the inverse iteration will lead to misconvergence in the nonlinear case. The residual inverse iteration proposed in [8] fixes this in the following way: according to the inverse iteration method we have:

$$\begin{aligned} x_k - x_{k+1} &= x_k + (\lambda_{k+1} - \lambda_k)T(\lambda_k)^{-1}T'(\lambda_k)x_k \\ &= T(\lambda_k)^{-1}(T(\lambda_k) + (\lambda_{k+1} - \lambda_k)T'(\lambda_k))x_k \\ &= T(\lambda_k)^{-1}T(\lambda_{k+1})x_k + \mathcal{O}(|\lambda_{k+1} - \lambda_k|^2) \end{aligned}$$

$$\Rightarrow x_{k+1} \approx x_k - T(\lambda_k)^{-1}T(\lambda_{k+1})x_k \tag{8}$$

where we can replace λ_k with the shift σ. Notice that in the third step we used a Taylor expansion of $T(\lambda_{k+1})$ around $T(\lambda_k)$.

The three important steps in the residual inverse iteration method are the following: first we solve λ_{k+1} from $v^*T(\sigma)^{-1}T(\lambda_{k+1})x_k = 0$, then we compute the residual $r_k = T(\lambda_{k+1})x_k$ and finally we solve $T(\sigma)d_k = r_k$ for d_k which we can use to find our next approximation x_{k+1}. Note that in this case we have to solve two systems per iteration instead of one as in the inverse iteration algorithm. However, since the system involves $T(\sigma)$ both times we only have to compute a decomposition once, before we start the algorithm.

3.3 Iterative Projection Methods

When discussing Arnoldi and Jacobi-Davidson type methods, we will follow the same path we did before: assuming that we have a search space of dimension m and an orthonormal basis V, we try to find a new approximation of the desired eigenvector and -value. To find the update, we use the basis of the search space to reduce the dimension of the original problem to m, that is we solve the nonlinear eigenproblem

$$V_k^H T(\vartheta_k)V_k y_k = 0.$$

If we have found a solution $(\theta_k, \tilde{y}_k)$ of this problem, and we write $x_k = V_k\tilde{y}_k$ then (θ_k, x_k) is a Ritz pair of T. We can use this Ritz pair to find an update for the approximated eigenpair. Also, we need these values to expand the search space. The way in which this is done makes the difference between Arnoldi and Jacobi-Davidson.

3.3.1 A General Algorithm for Nonlinear Eigenproblems

A general algorithm for projective methods is outlined as follows:

Algorithm 4 General projection method for nonlinear eigenproblems

```
start with an initial shift $\sigma$ and an orthonormal basis $V$
while $m \leq m_{max}$ do
  compute appropriate eigenvalue $\theta$ and corresponding eigenvector $y$ of the projected problem $V^H T(\lambda) V y = 0$
  determine Ritz vector $u = Vy$ and residual $r = T(\theta)u$
  if $||r||/||u|| < \epsilon$ then
    accept approximate eigenpair $\lambda_m = \theta$, $x_m = u$
    $m = m + 1$
    choose new shift $\sigma$
    restart if necessary
    determine approximations $\theta$ and $u$ of next wanted eigenvalue and vector
    compute residual $r = T(\theta)u$
  end if
  determine expansion of search space
end while
```

This algorithm is of course only a rough outline of an actual implementation. The main difficulty will be to determine a vector that expands the current search space, and this also the part where Arnoldi-type methods differ from Jacobi-Davidson.

3.3.2 Arnoldi type methods

It is not possible to extend the original Arnoldi method for standard eigenvalue problems to the nonlinear case because we can't construct a Krylov subspace to use as a search space. However, the method that is proposed by Voss in [17] and [18] is named after Arnoldi because the new search direction is orthonormalized against the previous vectors. The expansion of the current basis V_k can be chosen as $\hat{v}_{k+1} = x_k - T(\sigma)^{-1}T(\theta_k)x_k$ or equivalently as $v_{k+1} = T(\sigma)^{-1}T(\theta)x_k$ since x_k is the Ritz vector and therefore contained in V_k. This choice is based on the residual inverse iteration method, see expression 8. A drawback of this method is that we need to solve a large system because of the presence of $T(\sigma)^{-1}$ in the equation. When the problem under consideration is large (which is often the case) then solving will be too expensive. In this case the Arnoldi type method is only efficient if a reasonable preconditioner $M \approx T(\sigma)^{-1}$ is available.

3.3.3 Jacobi-Davidson type methods

A good alternative for Arnoldi is Jacobi-Davidson, especially when there is no preconditioner available. Again we assume to have a Ritz pair (θ_k, x_k). We can simply use the correction equation 2 to find an expansion of our basis:

$$\left(I - \frac{p_k x_k^*}{x_k^* p_k}\right) T(\theta_k) \left(I - \frac{x_k x_k^*}{x_k^* x_k}\right) z_{k+1} = -r_k,\ z_{k+1} \perp x_k$$

where we choose $p_k := T'(\theta_k)x_k$ and $r_k := T(\theta_k)x_k$. If we solve this equation approximately for z_{k+1} we can use this vector to expand V_k. Writing out the correction equation, and using that $x_k^* z_{k+1} = 0$ we find that:

$$T(\theta_k)z_{k+1} - \alpha_k p_k = -r_k,\ \text{with } \alpha_k := \frac{x_k^* T(\theta_k) z_{k+1}}{x_k^* p_k}$$

Solving for z_{k+1} gives that $z_{k+1} = -x_k + \alpha_k T(\theta_k)^{-1}T'(\theta_k)x_k$, and shows that $\tilde{z}_{k+1} := T(\theta_k)^{-1}T'(\theta_k)x_k \in \mathrm{span}[V_k, z_{k+1}]$. In [17] this vector $\tilde{z}_{k+1}$ is shown to be the direction in which the inverse iteration method with shift θ searches after one step. This means that just as in the linear case, Jacobi-Davidson is a subspace method based on shift-and-inverse iteration, giving quadratic convergence. Convergence may be even faster if the correction equation is preconditioned. See Sleijpen et al [13], section 7.1 for a discussion on preconditioning the correction equation.

4 Combustion

An important class of problems that often yields an eigenvalue problems is the class of acoustic problems. We are especially interested in thermo-acoustic modes in combustors. Much about this subject has been researched by Nicoud, Benoit, Sensiau and Poinsot in [9]. These modes are the main source of combustion oscillations. Because the acoustics in a gas chamber are coupled to the heat produced that comes with combustion, fluctuations in combustion may start to resonate according to their modes and grow over time. This will cause the combustor to become unstable. Until now, this behaviour was hard, if not impossible to predict by simulation during the design stage, because of a lack of sufficient computational power and the complex geometrical details of modern gas chambers. However, it is important to be able to predict and control these oscillations as early as possible.

4.1 Solution methods

Several very different methods have been proposed. Unfortunately, most methods are either too demanding in terms of computational work, or they are based on a greatly simplified model. An example of the first kind is to perform Large Eddy Simulations (LES), that are based on the full three-dimensional unsteady Navier Stokes equations. Another popular method is to model the geometry of the combustor, leaving out details, and to suppose the flame to be infinitely thin. In some cases, this method works good enough, but in general these assumptions are too restricting.

An intuitive method is to linearize the Navier-Stokes equations. The combustion oscillations have to be taken into account in the energy equation. If the heat release caused by the combustion oscillations can be modeled, the system of equations is closed. We can then proceed in two ways: we can solve the discretized PDE's in the time domain, or in the frequency domain using the Helmholtz equation instead of the wave equation. An important downside of the first option is that it only offers information about the most unstable modes. The second method yields an eigenvalue problem which will become non-linear when combustion occurs or when there is a boundary condition with impedance. In this study, we will follow the research from [9] of the second method, that is: to find a way of solving the Helmholtz equation arising from the thermo-acoustic problem numerically by writing it as a nonlinear eigenvalue problem. This is also the method used at CERFACS, where the second part of our study will be conducted.

4.2 Derivation of the equation

In this section, the acoustic problem will be described by the wave equation. Then, the equation will be transformed into a Helmholtz equation. The numerical solution methods in the following section will be aiming to approximate the solution of this equation.

4.2.1 Basic equations

Under certain assumptions, further specified in [9], we can describe the physical model by the following equations for mass density, momentum and entropy, together with the state equation and entropy expression:

$$
\begin{aligned}
\frac{D\rho}{Dt} &= -\rho\nabla\cdot u &(9)\\
\rho\frac{Du}{Dt} &= -\nabla p &(10)\\
\frac{Ds}{Dt} &= \frac{rq}{p} &(11)\\
\frac{p}{\rho} &= rT &(12)\\
s - s_{st} &= \int_{T_{st}}^{T} \frac{C_p(T')}{T'}dT' - rln\left(\frac{p}{p_{st}}\right) &(13)
\end{aligned}
$$

The variables that we introduced are: ρ is the density, u is the flow speed, p stands for the pressure, q is the heat release, s is the entropy and T is the temperature.

We linearize this by writing $p = p_0 + p_1$, $\rho = \rho_0 + \rho_1$ and $s = s_0 + s_1$. The second (fluctuating) term in these definitions is of order ϵ compared to the first (steady) term, where $\epsilon \ll 1$. In the linearization of u we assume that u_0/c_0, the Mach number, is practically zero. We write $u = u_1$ where $\sqrt{u_1 \cdot u_1}/c_0$ is also of order ϵ, where $c_0 = \sqrt{\gamma p_0/\rho_0}$ is the mean speed of sound, where γ is the heat capacity per unit mass at fixed pressure divided by the heat capacity per unit mass at fixed volume. With the assumptions of the zero Mach number and the neglection of the heat capacity fluctuations the following set of linear equations for the fluctuating quantities ρ_1, u_1, s_1 and p_1 is obtained:

$$
\begin{aligned}
\frac{\partial\rho_1}{\partial t} + u_1\cdot\nabla\rho_0 + \rho_0\nabla\cdot u_1 &= 0 &(14)\\
\rho_0\frac{\partial u_1}{\partial t} + \nabla p_1 &= 0 &(15)\\
\frac{\partial s_1}{\partial t} + u_1\cdot\nabla s_0 &= \frac{rq_1}{p_0} &(16)\\
\frac{p_1}{p_0} - \frac{\rho_1}{\rho_0} - \frac{T_1}{T_0} &= 0 &(17)\\
C_p\frac{T_1}{T_0} - rp_1p_0 &= s_1 &(18)
\end{aligned}
$$

In these equations, also the fluctating unknowns q_1 and T_1 play a role. To close the set of equations for the fluctuating quantities we need an equation that expresses q_1 in the other variables.

4.2.2 Flame response

Finding a suitable equation for q_1 is, from a physical point of view, the most difficult part of the modeling phase. A choice has to be made between working with global heat release from the whole flame zone or using a local flame model. For modern efficient combustors, the first model does not suffice since the flame is not acoustically compact, that is, the flame region is not small enough compared to the characteristic acoustic wavelength. The second model relates the local unsteady heat release to a reference acoustic velocity in the injector mouth. In equation form, we find that:

$$\frac{q_1(x,t)}{q_{tot}} = n_u(x)\frac{u_1(x_{ref}, t - \tau_u(x))\cdot n_{ref}}{U_{bulk}} \tag{19}$$

where $n_u(x)$ and $\tau_u(x)$ are fields of interaction index and time lag and n_{ref} is a fixed unitary vector defining the direction of the reference velocity. $n_u(x)$ has been made dimensionless by scaling u_1 and q_1 by U_{bulk} and q_{tot} respectively. The difficulty with this equation is that the fields of parameters $\tau_u(x)$ and $n_u(x)$ are hard to approximate empirically. The alternative to find reasonable values is to use compressible reacting LES.

By substituting q_1 in equation (16) using equation (19) we find the following equation:

$$\frac{\partial s_1}{\partial t} + u_1 \cdot \nabla s_0 = \frac{r}{p_0} n_u(x) \frac{q_{tot}}{U_{bulk}} u_1(x_{ref}, t - \tau_u(x)) \cdot n_{ref} \tag{20}$$

4.2.3 Helmholtz equation

Combining equations (14), (15), (17), (18) and (20) we arrive at the wave equation for p_1:

$$\nabla \cdot \left(\frac{1}{\rho_0} \nabla p_1 \right) - \frac{1}{\gamma p_0} \frac{\partial^2 p_1}{\partial t^2} = -\frac{\gamma - 1}{\gamma p_0} \frac{\partial q_1}{\partial t}. \tag{21}$$

Since this equation is linear in p_1, we can safely assume that all fluctuating variables will oscillate according to the same frequency $f = \omega/(2\pi)$. Therefore, we can introduce harmonic variations for pressure, velocity and local heat release perturbations:

$$p_1 = \mathcal{R}\rceil(\hat{p}(x)e^{-i\omega t}) \tag{22}$$

$$u_1 = \mathcal{R}\rceil(\hat{u}(x)e^{-i\omega t}) \tag{23}$$

$$q_1 = \mathcal{R}\rceil(\hat{q}(x)e^{-i\omega t}). \tag{24}$$

If we translate equations (19) and **??**ComEq8) into the frequency domain we have all the tools necessary to describe the transformed pressure field $\hat{p}$ by the following Helmholtz equation:

$$\nabla \cdot \left(\frac{1}{\rho_0} \nabla \hat{p} \right) - \frac{\omega^2}{\gamma p_0} \hat{p} = \frac{\gamma - 1}{\gamma p_0} \frac{q_{tot}}{\rho_0(x_{ref}) U_{bulk}} n_u(x) e^{i\omega \tau_u(x)} \nabla \hat{p}(x_{ref}) \cdot n_{ref} \tag{25}$$

4.2.4 Boundary conditions

An important part in modeling the thermo-acoustic behaviour in a gas chamber is the effect of the boundary conditions. Three different types of conditions are possible on the boundary $\partial\Omega$:

Zero pressure: This corresponds to boundaries that are fully reflective, and where the pressure should be equal to the outer pressure. This means that there can be no fluctuations, or in mathematical terms: $\hat{p} = 0$ on $\partial\Omega_D$

Zero normal velocity: This corresponds to boundaries where there can be no fluctuation in the velocity of the flow through the boundary. This happens at walls (where there is no flow at all) or at inlets where the velocity is supposed to be constant. Consequently, $\hat{u} \cdot n_{BC} = 0$. Combined with equation (15) this gives for the pressure: $\nabla \hat{p} \cdot n_{BC} = 0$ on $\partial\Omega_N$

Imposed reduced complex impedance: On boundaries where neither the pressure fluctuation nor the normal velocity fluctuation is zero, there will be a combination of both boundary conditions: $c_0 Z \nabla \hat{p} \cdot n_{BC} - i\omega\hat{p} = 0$ on $\partial\Omega_Z$ where Z is the imposed reduced complex impedance, that may depend on the frequency ω.

With this last boundary condition equation (25) will lead to a nonlinear eigenvalue problem. This will be further elaborated in the next section.

4.3 The numerical method

In this section, we will see how the Helmholtz equation (25) translates into a nonlinear eigenproblem. The ways to solve these problems have been described in Section 3. As discretizing the equation is not a simple task, we will first consider a simplified version of equation (25) where we don't take the flame response into account. That is, we will discretize the equation

$$\nabla \cdot \left(\frac{1}{\rho_0} \nabla \hat{p} \right) - \frac{\omega^2}{\gamma p_0} \hat{p} = 0$$

combined with the boundary conditions defined before.

4.3.1 Discretization of the equation

To discretize the above problem, we use the finite element method. We divide the domain Ω into tetrahedra and define a piecewise linear function ϕ_j for every vertex v_j. The testfunction ϕ_j equals 1 on it's respective vertex v_j and 0 on the other nodes, with linear interpolation. So, for every tetrahedron there are only four test functions that are not zero on the entire area. We can then approximate $\hat{p}$ by $\hat{p}(x) \approx \sum_j \hat{p}_j \phi_j(x)$, where $p_j = p(v_j)$. Because we already know that $\hat{p} = 0$ on $\partial\Omega_D$ from the boundary condition on $\partial\Omega_D$ we we can restrict ourself to the set of vertices S_v of the mesh which do not belong to $\partial\Omega_D$:

$$\hat{p}(x) \approx \sum_{j:v_j \in S_v} \hat{p}_j \phi_j(x).$$

There are now N unknowns, where N is the number of vertices belonging to $\Omega \setminus \partial\Omega_D$. Now the continuous function $\hat{p}$ is discretized: all that is left is to determine the complex coefficients p_j. This can be done by using the Galerkin method. Starting from the Helmholtz equation:

$$\nabla \cdot \left(\frac{1}{\rho_0} \nabla \hat{p} \right) - \frac{\omega^2}{\gamma p_0} \hat{p} = 0$$

we replace $\hat{p}$ by it's approximation, multiply with the test function and integrate over Ω, and obtain $\forall k \in S_v$:

$$\int_\Omega \phi_k \nabla \left(\frac{1}{\rho_0} \nabla \cdot \sum_{j:v_j \in S_v} \hat{p}_j \phi_j(x) \right) dx + \omega^2 \int_\Omega \frac{\phi_k}{\gamma p_0} \sum_{j:v_j \in S_v} \hat{p}_j \phi_j(x) dx = 0.$$

Interchanging the summation and integration operands, and taking out the constants p_j gives $\forall k \in S_v$:

$$\sum_{j:v_j \in S_v} \int_\Omega \frac{1}{\rho_0} \phi_k \nabla \cdot (\nabla \phi_j) dx \hat{p}_j + \omega \sum_{j:v_j \in S_v} \int_\Omega \frac{1}{\gamma p_0} \phi_k \phi_j dx \hat{p}_j = 0.$$

The first integral is integrated by parts:

$$\int \phi_k \nabla \cdot \nabla \phi_j dx = - \int \nabla \phi_k \nabla \phi_j dx + \oint \phi_k \nabla \phi_j \cdot n d\xi$$

So the equation becomes $\forall k \in S_v$:

$$\sum_{j:v_j \in S_v} \left(- \int_\Omega \frac{1}{\rho_0} \nabla \phi_k \nabla \phi_j dx + \int_{\partial\Omega} \frac{1}{\rho_0} \phi_k \nabla \phi_j \cdot n d\xi + \omega \int_\Omega \frac{1}{\gamma p_0} \phi_k \phi_j dx \right) \hat{p}_j = 0.$$

Note that $\phi_k(x) = 0 \; \forall k : v_k \in S_v, x \in \partial\Omega_D$ so $\partial\Omega_D$ does not contribute to the boundary integral. On $\partial\Omega_N$ we can rewrite the boundary integral by using the boundary condition $\nabla \hat{p} \cdot n = 0$:

$$\begin{array}{rcl}\sum_{j:v_j\in S_v}\int_{\partial\Omega}\frac{1}{\rho_0}\phi_k\nabla\phi_j\cdot ndx\hat{p}_j & = & \int_{\partial\Omega}\frac{1}{\rho_0}\phi_k\nabla(\sum_{j:v_j\in S_v}\phi_j\hat{p}_j)\cdot nd\xi \\ & = & \int_{\partial\Omega}\frac{1}{\rho_0}\phi_k\nabla\hat{p}\cdot nd\xi \\ & = & 0\end{array}.$$

Showing that only $\partial\Omega_Z$ contributes to the boundary integral. Rewriting the integral over $\partial\Omega_Z$ in the same way as for $\partial\Omega_N$ and substituting $\nabla\hat{p}\cdot n = i\omega/c_0Z\sum\phi_j\hat{p}$ gives us our final equation $\forall k \in S_v$:

$$\sum_{j:v_j\in S_v}\left(-\int_\Omega\frac{1}{\rho_0}\nabla\phi_k\nabla\phi_j dx + \int_{\partial\Omega_Z}\frac{1}{\rho_0c_0Z}\phi_k\phi_j d\xi + \omega\int_\Omega\frac{1}{\gamma p_0}\phi_k\phi_j dx\right)\hat{p}_j = 0.$$

Combining all N equations for all N unknowns p_j into a matrix equation gives:

$$AP + \omega B(\omega)P + \omega^2 CP = 0 \qquad (26)$$

where P is the vector containing the unknowns p_j and A, B and C are symmetric matrices with generic element

$$\begin{aligned}A_{kj} &= -\int_\Omega\frac{1}{\rho_0}\nabla\phi_k\nabla\phi_j dx \\ B_{kj} &= \int_{\partial\Omega_Z}\frac{1}{\rho_0c_0Z}\phi_k\phi_j d\xi \\ C_{kj} &= \int_\Omega\frac{1}{\gamma p_0}\phi_k\phi_j dx.\end{aligned}$$

To get rid of the nonlinearity that is caused by the fact that B depends on Z and therefore on ω Nicoud et al. suggest to model the impedance by $1/Z = 1/Z_0 + Z_1\omega + Z_2/\omega$ so because of the multiplication by ω it will be possible to rewrite the problem as a quadratic problem. This makes it easier to solve with traditional methods, but is not necessary since efficient solution methods have been developed, as stressed in section 3.

4.3.2 Incorporating the flame response

In the context of the finite elements method described above it is not at all hard to incorporate flame response. Equation (25) needs to be discretized completely, whereas we only did two terms in the section before. So all that is left is to discretize is $\frac{\gamma-1}{\gamma p_0}\frac{q_{tot}}{\rho_0(x_{ref})U_{bulk}}n_u(x)e^{i\omega\tau_u(x)}\nabla\hat{p}(x_{ref})\cdot n_{ref}$ in terms of $\hat{p}_j$. Following the same strategy as before, this term gives us another term DP in the matrix equation (26) with generic element

$$D_{kj} = \int_\Omega\frac{\gamma-1}{\gamma}\frac{q_{tot}}{\rho_0(x_{ref})}U_{bulk}n_u(x)\exp^{i\omega\tau_u(x)}\phi_k\nabla\phi_j(x_{ref})\cdot n_{ref}dx.$$

As is noted in [9], this term is nonlinear in ω and cannot be as easily be rewritten in such a way that we are provided with a quadratic eigenvalue problem, as can be done for the nonlinearity of the impedance. So, we must find a way to solve the following nonlinear eigenvalue problem:

$$(A - D(\omega))P + \omega B(\omega)P + \omega^2 CP = 0.$$

Nicoud et al suggest to use a simple Picard iteration:

$$(A - D(\omega_k - 1))P + \omega_k B(\omega_{k-1})P + \omega_k^2 CP = 0,$$

or with the assumption about the impedance that $1/Z = 1/Z_0 + Z_1\omega + Z_2/\omega$

$$(\mathcal{A} - D(\omega_k - 1))P + \omega_k\mathcal{B}P + \omega_k^2\mathcal{C}P = 0$$

where $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ are altered versions of A, B and C to take the modelled impedance in account. However, with the modern techniques described in 3 this is not necessary, we can solve the nonlinear eigenproblems without rewriting them as linear problems.

5 Research Questions

We have studied solution methods of various types of eigenvalue problems, culminating in a general subspace method for nonlinear eigenproblems. This general method can be implemented to resemble Arnoldi, or we can use the correction equation suggested by the Jacobi-Davidson method. A third possibility is to linearize the nonlinear eigenproblem, for example by a Picard iteration where a known approximation for the eigenvalue is inserted in the nonlinear term to obtain a quadratic eigenvalue problem, which can be rewritten as a standard eigenvalue problem and solved with the original Arnoldi method.

Our goal is to implement Jacobi-Davidson for nonlinear eigenvalue problems, to compare it with the Arnoldi-type Picard-iteration. This will be done through a couple of testproblems with known eigenvalues. Also, we will try to find improvements for the Jacobi-Davidson method in the form of preconditioning, choice of solution method for the small eigenvalue problem, choice of iterative method for approximation of the solution of the correction equation and other features.

We will also use the results of our theoretical study to improve the current methods used at CERFACS to solve the application to realistic combustion problems. We will do this in first instance by using the same Picard-iteration as is currently done, and then improving the method that is used to solve the remaining quadratic problem. Finally we will try to use Jacobi-Davidson for the full non-linear problem.

References

[1] W.E. Arnoldi, *The principle of minimized iteration in the solution of the matrix eigenvalue problem*, Quarterly of Applied Mathematics, volume 9, pages 17-25 (1951).

[2] E.R. Davidson, *The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices*, Journal of Computational Physics, volume 17, pages 87-94 (1975).

[3] G.H. Golub and C.F. van Loan, **Matrix Computations**, The Johns Hopkins University Press, Baltimore and Londen (1983).

[4] C.G.J. Jacobi, *ˋ´Uber ein leichtes Verfahren, die in der Theorie der Skularstrungen vorkommenden Gleichungen numerisch aufzulsen* (1846).

[5] C. Lancszos, *An iteration method for the solution of the eigenvalue problem of the linear differential and integral operators*, Journal on Res. Nat. Bur. Stand., 45 pages 51-94 (1950).

[6] R.B. Lehoucq, D.C. Sorensen and C. Yang, *ARPACK Users' Guide: Solution of Large Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*, SIAM, Philadelphia, (1998).

[7] V. Mehrmann and H. Voss, *Nonlinear Eigenvalue Problems: A challenge for Modern Eigenvalue Methods*, Report 83, Arbeitsbereich Mathematik, TU Hamburg-Harburg (2004) GAMM Mitteilungen 27, pages 121-152 (2004).

[8] A. Neumaier, *Residual inverse iteration for the nonlinear eigenvalue problem*, SIAM Journal on Numerical Analysis, Volume 22, pages 914-923 (1985).

[9] F. Nicoud, L. Benoit, C. Sensiau and T. Poinsot, *Computation of 3D thermo-acoustic modes in combustors including comlex impedances and multidimensional flames*, in preparation.

[10] C.C. Paige, B.N. Parlett, and H.A. van der Vorst, *Approximate solutions and eigenvalue bounds from Krylov subspaces*, Numerical Linear Algebra Applications, Volume 2, pages 115-134 (1995).

[11] Y. Saad, **Numerical Methods for Large Eigenvalue Problems** (1992), Manchester University Press Series in Algorithms and Architectures for Advanced Scientific Computing.

[12] Y. Saad, Martin H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM Journal on Scientific and Statistical Computing, Volume 7, pages 856-869. Society for Industrial and Applied Mathematics, Philadelphia (1986).

[13] G.L.G. Sleijpen, J.G.L. Booten, D.R. Fokkema and H.A. van der Vorst, *Jacobi-Davidson type methods for generalized eigenproblems and polynomial eigenproblems: Part I*, Preprint 923, Department of Mathematics, University Utrecht (September, 1995. Revised, November, 1995).

[14] G.L.G. Sleijpen and H.A. van der Vorst, *A Jacobi-Davidson Iteration Method for Linear Eigenvalue Problems* SIAM Journal on Matrix Analysis and Applications, Volume 17, Number 2, pages 401-425 (1996).

[15] G.L.G. Sleijpen, H.A. van der Vorst and Zhaojun Bai *Jacobi-Davidson algorithms for various eigenproblems -A working document-*, Preprint nr. 1114 Department of Mathematics, University Utrecht, (1999).

[16] H. Voss, *Numerical Methods for Sparse Nonlinear Eigenvalue Problems*, Report 70, Arbeitsbereich Mathematik, TU Hamburg-Harburg (2004) Proc. XVth Summer School on Software and Algorithms of Numerical Mathematics, Hejnice, Czech Republic

[17] H. Voss *An Arnoldi method for nonlinear symmetric eigenvalue problems*, Online Proceedings of the SIAM Conference on Applied Linear Algebra, Williamsburg, http://www.siam.org/meetings/laa03/ (2003).

[18] H. Voss *An Arnoldi method for nonlinear eigenvalue problems* BIT Numerical Mathematics 44, pages 387-401 (2004).