Chapter 2

An optimal adaptive methodology for the resolution of the ROM equation

A new algorithm (A_{RL}), intended to enclose the roots of thermo-acoustic characteristic equations more efficiently is reported in this chapter. It can solve numerically, in a robust and efficient way, the various existing ROM equations. Most of the former thermo-acoustic studies relied on simple algorithms such as the Newton-Raphson method (see chapter 1 and the A_{NR} algorithm), to find the complex-valued frequencies of the acoustic modes one by one, or Nyquist diagram analysis to predict stability. However, the method described here, based on the argument principle, is built specifically to retrieve all the solutions of a ROM. Several solvers based on this principle have already been reported [1, 2, 3] and exhibited robust properties. However, they all suffered from expensive computational costs. Here, the A_{RL} algorithm comes with two major improvements. First, after a short comparison with previous solvers, the best ideas of each are gathered in the A_{NR} solver so that the computational cost is lowered. Second, a new method is employed to enclose multiple zeros (whatever the order of multiplicity) at a negligible cost. Finally, the implementation of the A_{RL} algorithm exhibits very good performance in term of computational time when compared to other algorithms found in the literature.

Introduction

In chapter 1, we derived a simple ROM for the INTRIG burner and showed that algorithms only based on Newton-Raphson (NR) methods may be used to find some solutions of equation 1.6.5 (Hopefully, all of them). However, the whole set of solutions must be computed to assess the stability of the thermo-acoustic system.

It can be seen that equation 1.6.5 is highly non-linear since it involves multiple exponentials terms. Moreover, f_0 derives from a certain class of functions of the complex variable z or ω that are called *meromorphic functions*, for which the numbers and values of the poles and zeros can be obtained with contour integrals. In the following, we will restrict our target to the case of *holomorphic functions*, a restriction of meromorphic functions for which there is no pole in the entire domain of interest. This assumption holds for all the ROM equations presented in this manuscript. However, in the case of a function containing some poles, the domain of research can be adapted to exclude them as done in [4]. This restriction also implies that all models or fit used in ROM such as flame transfer functions or transfer matrices coefficients are also holomorphic.

The objective of this study is to find the roots of a given analytic function $f : \mathbb{C} \to \mathbb{C}$ in a bounded domain of the complex plane. To solve this problem, an efficient and robust algorithm has been designed and implemented with success. The backbone of the algorithm stems from a generalized version of *Cauchy's argument principle*, as first described by Delves and Lyness [1, 5]. This theorem is used recursively throughout an adaptive multilevel subdivision strategy, the challenge being to find a satisfactory compromise between robustness and computational efficiency.

Algorithms based on the principle argument have been detailed and implemented more recently by Dellnitz [2]. Johnson et al [3, 4] proposed two implementations which used validated integral computations. However, theses methods, albeit mathematically rigorous, happen to be computationally intensive. Other families of algorithm that do not rely explicitly on *Cauchy's argument principle* have been developed. Suzuki et al [6] proposed a method which relies on the Numerical Integration Error Method derivation. Ko et al [7] first designed a method to compute the topological degree of a mapping in $\mathbb{R}^2$ and then applied these results to analytic functions in $\mathbb{C}$. Kopitz et al [8] resolved thermo-acoustic ROM by using the Nyquist criterion, which stems from Cauchy's argument principle. Finally, Semenov[9] described a method which only work with non multiple zeros. The A_{RL} algorithm combines some of the advantages of the previous solvers, in addition to some new extensions dedicated to enhance its robustness and rapidity.

In section 1.1, we recall the theoretical background associated with analytic functions, and state the generalized *Cauchy's argument principle.* The algorithm is detailed in section 1.2. Sections 1.3 and 1.4 elaborate on the particular implementation of the algorithm. Section 1.5 displays results and comparison with other solvers in terms of accuracy, robustness and efficiency on both standard performance benchmarks and thermo-acoustic ROM. Finally, a parametric study concerning the INTRIG burner is realized (section 1.6).

2.1 Theoretical context

The resolution of a characteristic equation in a compact subset $\mathbf{U}$ of the complex plane like equation 1.6.5 is equivalent to the following problem:

$$\{z \in \mathbf{U} \,|\, f(z) = 0\} \tag{2.1.1}$$

2.1.1 Cauchy's argument principle

The argument principle, based on an integration over a closed contour $\partial\mathbf{U}$, allows to retrieve the number of roots of a holomorphic function contained within $\partial\mathbf{U}$. Let $f : z \in \mathbf{U} \rightarrow f(z) \in \mathbb{C}$ be a holomorphic, non constant function and $\mathbf{U} \in \mathbb{C}$ is defined by its closed contour $\partial\mathbf{U}$. As f is analytical within $\mathbf{U}$, it is equal to its Taylor series [1]:

$$f(z) = \sum_{k=0}^{\infty} a_k z^k \,,\; z \in \mathbf{U} \tag{2.1.2}$$

which can be recast into a product form:

$$f(z) = g(z) \prod_{i=1}^{n} (z - z_i)^{p_i} \tag{2.1.3}$$

where z_i, $i \in [1, n]$ are the zeros of f contained in $\mathbf{U}$, $p_i \geq 0$ their multiplicity and $g : z \in \mathbf{U} \rightarrow g(z) \in \mathbb{C}$ is an holomorphic function containing no zeros in $\mathbf{U}$ nor in $\partial\mathbf{U}$. The ratio between f

[1] This equality is the definition of an analytic function and holds for most usual functions.

and its derivative f' is equal to:

$$\frac{f'(z)}{f(z)} = \sum_{i=1}^{n} \frac{p_i}{z - z_i} + \frac{g'(z)}{g(z)} \tag{2.1.4}$$

which can be integrated over the closed contour $\partial\mathbf{U}$:

$$\frac{1}{2\pi i} \oint_{\partial\mathbf{U}} \frac{f'(z)}{f(z)} dz = \sum_{i=1}^{n} \frac{1}{2\pi i} \oint_{\partial\mathbf{U}} \frac{p_i}{z - z_i} dz + \frac{1}{2\pi i} \oint_{\partial\mathbf{U}} \frac{g'(z)}{g(z)} dz \tag{2.1.5}$$

This expression can be simplified by using both Cauchy's integral theorem (equation 1.1.7) and formula (equation 1.1.7) which are valid for holomorphic functions within $\mathbf{U}$:

$$\frac{1}{2\pi i} \oint_{\partial\mathbf{U}} h(z) dz = 0 \tag{2.1.6}$$

$$\frac{1}{2\pi i} \oint_{\partial\mathbf{U}} \frac{h(z)}{z - a} dz = h(a)\delta(a, \mathbf{U}) \tag{2.1.7}$$

where $\delta(a, \mathbf{U}) = 1$ if a lies inside $\mathbf{U}$, 0 else. The second term of the RHS of equation 1.1.5 is equal to 0 as $g'(z)/g(z)$ is still holomorphic in $\mathbf{U}$. The first term of the RHS is equal to $N_{\text{sol}} = \sum_{i=1}^{n} p_i$. Finally, one obtains the Cauchy's argument principle:

$$I_0(\partial\mathbf{U}) = \frac{1}{2\pi i} \oint_{\partial\mathbf{U}} \frac{f'(z)}{f(z)} dz = N_{\text{sol}} \tag{2.1.8}$$

where N_{sol} is the number of zeros of f contained in $\mathbf{U}$.

If the tested function is only meromorphic, so that is has poles, one gets $I_0(\partial\mathbf{U}) = N_{\text{sol}} - N_{\text{poles}}$ and it is thus not possible to retrieve the number of zeros contained in $\mathbf{U}$.

2.1.2 Extension of the argument principle

Results provided by equation 1.1.8 can help to enclose the roots of holomorphic functions: knowing the number of zeros of f is the first step of an algorithm dedicated to find all the solutions of equation 1.1.1. It is however possible to go further with the generalized argument principle [1, 5]:

$$I_k(\partial\mathbf{U}) = \frac{1}{2i\pi} \oint_{\partial\mathbf{U}} z^k \frac{f'(z)}{f(z)} \mathrm{d}z = s_{\mathrm{k}} \tag{2.1.9}$$

$$s_{\mathrm{k}} = \sum_{i=1}^{n} p_i z_i^k \tag{2.1.10}$$

where $\{z_i, p_i\}$ are the zeros of f and their respective multiplicity. With $k = 0$, one retrieves the number of zeros contained in $\mathbf{U}$. The sum of all these roots is obtained for $k = 1$. In the case of a domain $\mathbf{U}$ containing only one root z_1, the latter can be directly extracted as $z_1 = I_1(\partial\mathbf{U})$. In the general case, once the number of zeros contained within $\partial\mathbf{U}$ is known, it is possible to construct an equivalent polynomial $Q_{\mathrm{N}}(z)$ of order $N = N_{\text{sol}}$ which coefficients q_k are directly linked with the power sums s_{k} so that the zeros of Q_{N} correspond to the zeros of f contained in $\mathbf{U}$ [1]. The method used to obtain such a polynomial is detailed in section 1.1.3.

2.1.3 Construction of the equivalent polynomial

Newton's Identities provide the link between the power sums s_k and the corresponding equivalent polynomial Q_N [10] :

$$Q_\mathrm{N}(z) = \sum_{j=0}^{N} (-1)^j e_j z^{N-j} \tag{2.1.11}$$

$$\text{with } e_0 = 1\,,\; e_{k\neq 0} = \frac{1}{k}\sum_{j=1}^{k}(-1)^{j+1} e_{k-j} s_j$$

The polynomial Q_N is constructed to share the same roots as f in $\mathbf{U}$. Analytical methods exist to solve this polynomial up to the fourth order ($N = 4$). In the $N = 3$ case, the following equivalent polynomial is obtained:

$$Q_3(z) = z^3 - s_1 z^2 + \frac{s_1^2 - s_2}{2} z - \frac{1}{6}(2s_3 - 3s_1 s_2 + s_1^3) \tag{2.1.12}$$

2.2 Algorithm description

The A_RL algorithm dedicated to compute all the solutions of equation 1.1.1 within a closed subset $\mathbf{U} \subset \mathbb{C}$ is now detailed. An example is then provided to illustrate its behaviour.

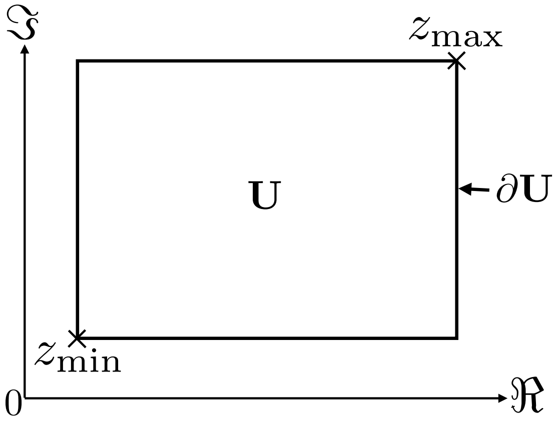

Figure 2.1: Initial mesh $\mathbf{U} \subset \mathbb{C}$ defined by $Z_\mathrm{min}, Z_\mathrm{max}$.

The domain $\mathbf{U}$, which has a rectangular shape, is defined by:

$$\begin{aligned}\mathbf{U} = \{z \in \mathbb{C} \,|\, \{&\Re(Z_\mathrm{min}) \leq \Re(z) \leq \Re(Z_\mathrm{max})\,,\\ &\Im(Z_\mathrm{min}) \leq \Im(z) \leq \Im(Z_\mathrm{max})\}\end{aligned} \tag{2.2.1}$$

This domain, shown in Fig. 1.1, is called the initial mesh and its contour is called $\partial\mathbf{U}_\mathrm{mesh}$. The A_RL algorithm is applied recursively to compute all the solutions of equation 1.1.1: the number of root is identified with the Cauchy's argument and the problem is locally recast into an equivalent polynomial resolution.

A_{RL} algorithm

1. Generalized Cauchy's argument principle (equation 1.1.9) is used to compute both the number of roots N ($k = 0$) which are contained in the current mesh and the corresponding power sums ($k > 0$). If a root is located too close to the boundary $\partial \mathbf{U}_{\mathrm{mesh}}$, all the adjacent meshes are reshaped. This procedure is detailed in section 1.3. Once the number of root is known, the problem is reformulated in term of an equivalent polynomial, which maximal order is limited to $N_{\max}$.

2. **If** $N = 0$

   There are no roots in this mesh, nothing to be done.

   If $1 \leq N \leq N_{\max}$

   The solutions are obtained in three steps:

   (a) The equivalent polynomial Q_{N} is constructed according to Eq. 1.1.11.

   (b) The equivalent polynomial Q_{N} is solved with an appropriate analytical method. The roots of Q_{N} are only approximations of the roots of f as its coefficients are provided by non exact, numerical integration.

   (c) Starting from the roots of the equivalent polynomial, the NR algorithm is applied to compute the zeros of f with a better accuracy. All zeros are finally strictly checked with an integral computation : this procedure is described in the appendix **??**.

   If $N > N_{\max}$

   The existence of a single root of multiplicity N is first checked (see section 1.3). If not found, the mesh is split in two sub-meshes and the procedure described in step 1 is applied on both of them.

3. If the refinement reaches a user-defined size $\mathbf{L} = |Z_{\max}(\mathbf{U}_{\mathrm{mesh}}) - Z_{\min}(\mathbf{U}_{\mathrm{mesh}})| < \epsilon$ and there are still more than $N_{\max}$ roots within this mesh, the refinement is stopped and a single root (of multiplicity N) is extracted and set to $(Z_{\min}(\mathbf{U}_{\mathrm{mesh}}) + Z_{\max}(\mathbf{U}_{\mathrm{mesh}}))/2$.

To illustrate this algorithm, an example is described in Fig. 1.2. Here, the maximum equivalent polynomial order is set to $N_{\max} = 2$ for the sake of clarity. During the first step, four solutions are found in the initial mesh M_1. As $4 > N_{\max}$, the latter is split in two sub-meshes: M_3 on the left and M_2 on the right. Only one root is found in M_2 and is enclosed by first, the resolution of the first order equivalent polynomial and then by an additional use of the NR method. Meanwhile, M_3 still contains $3 > N_{\max}$ roots and must be split. At step 3, a root lies exactly on the boundary so that the computations of the number of solutions failed in both lower and upper sub-meshes. As a consequence, both of them are reshaped. Finally, two solutions are extracted from the upper mesh M_4 by solving the equivalent polynomial and one is extracted from the lower mesh M_5.

2.3 Strategies of domain splitting and reshaping

When $N > N_{\max}$ in a given mesh, a two-step checking procedure is performed to decide whether a single root of multiplicity N is located in it or not. Assuming a zero z_{N} of multiplicity N in equation 1.1.10 leads to

$$s_2 - \frac{s_1^2}{N} = N z_{\mathrm{N}}^2 - \frac{(N z_{\mathrm{N}})^2}{N} = 0 \qquad (2.3.1)$$

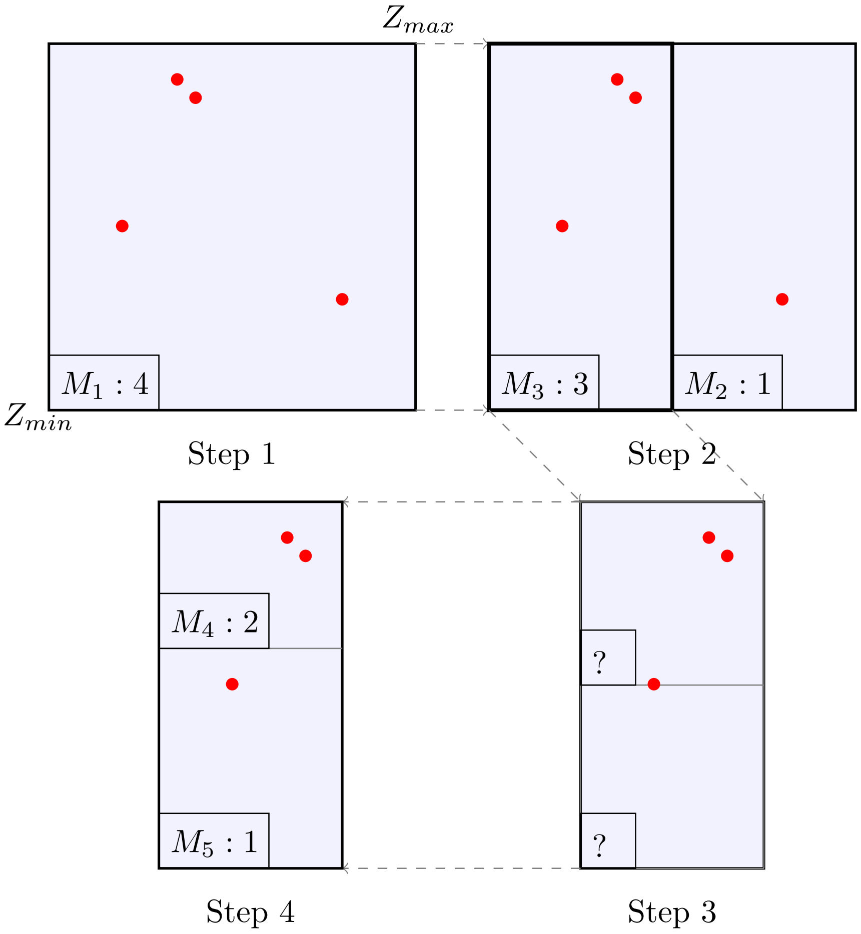

Figure 2.2: Illustration of the algorithm used to find the solutions of equation $f(z) = 0$. Step 1: initial domain of research. Step 2: creation of the sub-meshes. Step 3: problem of convergence for both lower and upper sub-meshes. Step 4 : sub-meshes are reshaped.

As s_1 and s_2 are not computed exactly, the following condition is tested to detect the existence of a multiple root:

$$d = |\frac{s_2}{N} - \left(\frac{s_1}{N}\right)^2|^{\frac{1}{2}} < 0.001L_{\text{mesh}} \tag{2.3.2}$$

As long as s_1 and s_2 are known with a sufficient precision, equation 1.3.2 should be verified for a single root of multiplicity N. However, this criterion is not sufficient to ensure the existence of the single root as it may be fulfilled with distinct roots too. For instance, the three single roots of equation $z^3 - 1 = 0$: $\{1\,,\, e^{i2\pi/3}\,,\, e^{i4\pi/3}\}$, lead to $s_1 = 0$ and $s_2 = 0$ so that $d = 0 < 0.001L_{\text{mesh}}$.

In the second step, an approximation of the root is provided by $z_{\text{N}} = s_1/N$. The NR method applied to the function $f^{1/N}$, starting from this guess. In case of convergence, the existence of the single root is proved with contour integration.

If the two-step checking procedure fails, the mesh is split as it contains too many distinct roots. Various mesh shapes have been used in previous studies: rectangles [3, 7], triangles [4] or even circles [1], for which there is overlapping. The rectangular shape was preferred in this study (Fig. 1.2). Meshes can be split along the real or the imaginary axis. Here, a simple solution is retained: the mesh is cut in its larger dimension as done in [3]. Once split into two sub-meshes, some roots may lie too close to the shared boundary. Such situations are identified by performing a new contour integration:

$$I_\infty(\partial\mathbf{U}) = \frac{1}{2\pi i}\oint_{\partial\mathbf{U}} \left|\frac{f'(z)}{f(z)}\right| \mathrm{d}z \tag{2.3.3}$$

which numerically diverges if a root of whatever multiplicity is located within the boundary. In the vicinity of the root z_i of multipliticy p_i, equation 1.1.4 shows that f'/f is equivalent to:

$$\left|\frac{f'}{f}(z)\right| \approx \frac{p_i}{|z - z_i|} \tag{2.3.4}$$

which is not integrable on paths containing z_i [2]. Except for the initial domain, specified by the user, these roots are always located on the common boundary of the two meshes issued from the last split. If divergence occurs in the initial mesh, a domain sightly larger is used instead and only the roots contained in the initial domain are kept. If the divergence occurs in a sub-mesh, the shared boundary is translated and a new computation is performed. An example is given in Fig. 1.2 between steps 3 and 4.

2.4 Numerical scheme used for the computation of the generalized argument principle

The computation of the generalized argument principle requires $4N_{\max}$ integrations for each mesh. Each integral requires a line integration in the complex plane:

$$\begin{aligned}\Gamma_k &= \frac{1}{2i\pi}\int_{z_1}^{z_2} g_k(z)\mathrm{d}z\,,\ 0 \le k \le N_{\max} \\ g_k(z) &= z^k \frac{f'(z)}{f(z)}\end{aligned} \tag{2.4.1}$$

All Γ_k integrals on the same path $\mathbf{P} = [z_1 \to z_2]$ are computed simultaneously to avoid redundant calls to f and its derivative. Three integration schemes have been implemented in this study. The first one, called the Modified Romberg scheme (MR), is an adaptive scheme based on a fourth order approximation of the derivative f'. The two others are the well known Romberg scheme (R) and Adaptive Simpson scheme (AS), which both require automated differentiation for f'. All these schemes must return a result with a controlled accuracy $\Gamma_{\text{err}} < 1/8$ in order to estimate properly $I_0(\partial\mathbf{U})$. In practice, a criteria of $\Gamma_{\text{err}} = 0.01$ was preferred as the integration error is only estimated.

2.4.1 Modified Romberg scheme (MR)

To avoid the use of automatic differentiation, the MR scheme, derived from the Romberg adaptive [11, 2] scheme was implemented and tested. The composite Simpson's rule is first applied with $n = 2$ initial subintervals to compute $\Gamma^{\text{MR}}_{k,2}$. The same rule is then applied recursively with twice more subintervals at each iteration to compute $\Gamma^{\text{MR}}_{k,2^i}$. At each step $i > 1$, the integration error is approximated by $\Gamma_{\text{err}} = |\Gamma^{\text{MR}}_{k,2^i} - \Gamma^{\text{MR}}_{k,2^{i-1}}|$ and the computation stops when it is sufficiently low. All the g_k computed during step i are reused at step $i+1$ so that the function f is called as few as possible.

In order to compute $g_k(z) = z^k \frac{f'(z)}{f(z)}$, a fourth-order finite difference centred stencil is used to evaluate $f'(z)$ so that additional calls to f are needed to compute the derivatives at the extremities of the path $\mathbf{P}$. However, this overhead only add two additional calls to f during each iteration as shown in figure 1.3.

[2] This complex integral can be recast into $\int_{-1}^{1} |t|^{-1} dt$ which do not converge

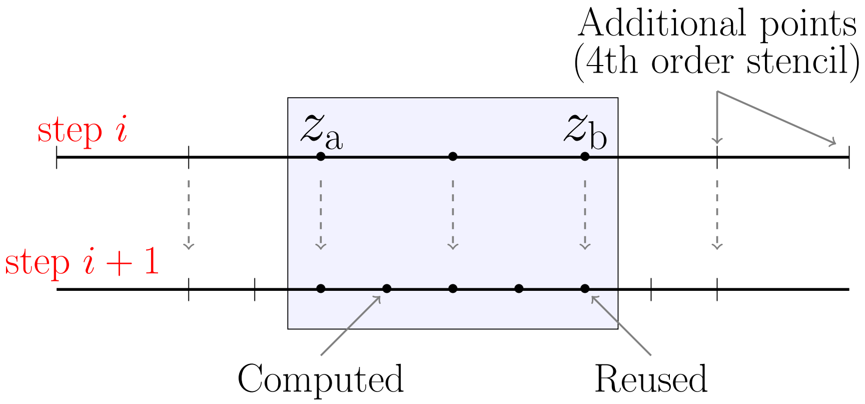

Figure 2.3: Illustration of the MR scheme of integration. During step $i + 1$, values computed from step i are reused. Additional values are computed to compute the derivative with a fourth order finite difference centred stencil.

2.4.2 Romberg scheme with automated differentiation (R)

The R scheme [11] has been implemented and tested. By using automated differentiation, the $g_k(z_i)$ are exactly computed so that truncation errors are avoided. Contrary to the MR scheme which was limited to fourth order accuracy, the R scheme's order of accuracy is increasing with the number of iterations. This scheme provides better performance than the MR scheme when the function f is simple. However, it may become slower when the extra computational time necessary to compute the derivative of f increases.

2.4.3 Adaptive Simpson Scheme with automated differentiation (AS)

Finally, the AS scheme [3] was also tested. In this scheme, the fourth order Simpson rule of integration is applied in any subdivision $\mathbf{P}_{\mathrm{ab}}$ of the original path. These subdivisions are split in two parts: $\{\mathbf{P}_{\mathrm{ac}}, \mathbf{P}_{\mathrm{cb}}\}$, only when more precision is needed according to the following relation [12]:

$$|\Gamma_0^{\mathrm{AS}}(\mathbf{P}_{\mathrm{ac}}) + \Gamma_0^{\mathrm{AS}}(\mathbf{P}_{\mathrm{cb}}) - \Gamma_0^{\mathrm{AS}}(\mathbf{P}_{\mathrm{ab}})| > 15\frac{\Gamma_{\mathrm{err}}}{2^{\mathrm{Nsplit}}} \tag{2.4.2}$$

where $1/2^{\mathrm{Nsplit}}$ refers to the relative length of the newly created subdivision compared to the original path. An illustration of this adaptive integration scheme is depicted in Fig. 1.4.

For performance purposes, the convergence criteria is only checked for $k = 0$ as only the computation of the number of roots in a mesh must be carried out in a safe manner. Finally, a global fifth-order of accuracy is obtained by combining the last two results obtained for each subdivisions. This scheme exhibits very good performance when compared to MR and R schemes as less calls to f are needed to provide the same precision. Indeed, far from the zeros of f, the g_k functions are rather smooth so that a uniform refinement as proposed in both R and MR schemes is not well suited. A quantitative comparison of all schemes is given in section 1.5.

2.5 Numerical results

The implementation of the A_{RL} algorithm has been realized in Java, into a program called RootLocker. The ANTLR library [13] is used to translate the equation $f(z) = 0$ proposed by the user into Java code which is then dynamically compiled and loaded by the Java virtual machine. This technique has several

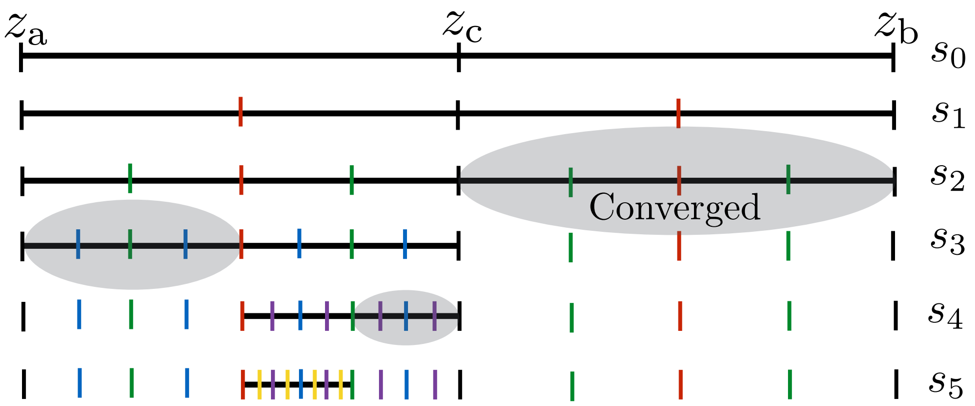

Figure 2.4: Illustration of the AS scheme of integration. All subdivisions are converged after step s_5 so that function f has been called 21 times

advantages. It first simplifies the implementation of the forward mode automatic differentiation [14] by using the source code transformation technique. Moreover, it allows faster computation by using only native Java code to describe the function f and its derivative. Finally, this procedure is transparent for the user, who only needs to describe the function f.

In the following part of this section, a few examples illustrating the improvement realized in terms of accuracy and performance compared to previous solvers are detailed. Example 1 is taken from [3], example 2 is adapted from [7] and example 3 is taken from [4]. The last example focus on a thermo-acoustic model proposed in [15]. Examples 1, 2 and 3 were all carried out with $N_{\text{max}} = 3$ while the influence of this parameter is studied in example 4. In all cases, two performance measurements are provided. First, the total number of calls to the function f is monitored. However, these data are not always available in all the reference papers for the first three examples and thus, cannot be compared. As a consequence, both the number of meshes and the computational time measured on a macbook air 2013 are provided (these data are available in the aforementioned references).

2.5.1 Academic cases

The first two examples concern complex polynomial equations.
Example 1:

$$f_1(z) = z^{11} - a \quad , \quad a = \frac{1}{2} + i\frac{\sqrt{3}}{2} \tag{2.5.1}$$

This polynomial has exactly 11 zeros which are uniformely distributed on the unit circle as shown in Fig. 1.5. The computation was conducted in the initial domain $\mathbf{U} = [Z_{\text{min}} = -3 - 3i, Z_{\text{max}} = 3 + 3i]$ and required 1270 calls to f_1 to compute the roots with seven decimals with the AS scheme. This corresponds to a computational time of $t = 5\ ms$ to be compared with the 40 seconds obtained in [3]. All other schemes of integration provided exactly the same roots, but with a longer computational time.

An other computation was realised with $a = 1$. In this case, one root is located on the real axis thus two sub-meshes needed reshaping (see Fig. 1.5, right). Despite this difficulty, only 1461 calls to $f1$ were needed to compute all the roots (corresponding to a time $t = 6$ ms). Finally, the criterion defined in equation 1.3.2 was verified in the initial mesh for both values of a. Checking that this was not caused by the presence of a single root of multiplicity $N = 11$ only required two additional calls to $f1$ as the NR algorithm diverged rapidly out of the domain.

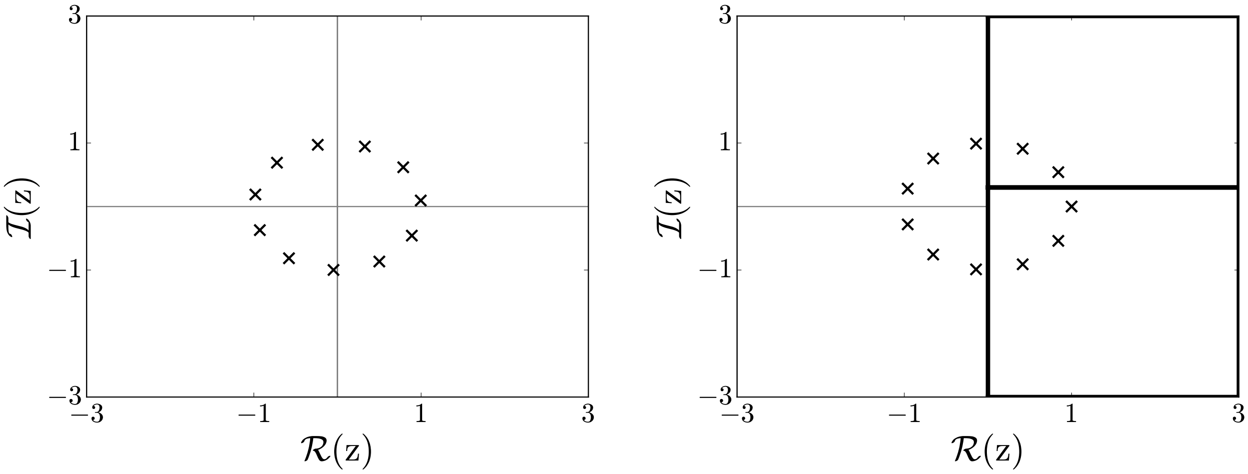

Figure 2.5: Location of the roots of f_1 for $a = 1/2 + i\sqrt{3}/2$ (left) and $a = 1$ (right). Meshes are represented in thin grey lines except for reshaped ones, which are represented in thick black lines.

Example 2:

$$f_2(z) = (z^2 + z + 1)^2(z - 1)^4(z^3 + z^2 + z + 1)^3(z - 2)(z - 4)^4 \tag{2.5.2}$$

This polynomial has exactly 22 roots (see Fig. 1.7), all located in the domain of research defined by $\mathbf{U} = [Z_{\min} = -5 - 5i, Z_{\max} = 5 + 5i]$. This example exhibits fourth order roots which cannot be computed directly with $N_{\max} = 3$. For this reason, they are enclosed thanks to the criteria defined in equation 1.3.2. Finally, 4564 calls to f_2 were needed with the AS scheme. This corresponds to a computational time of $t = 26$ ms which may be compared with the 22.7 seconds obtained in [7].

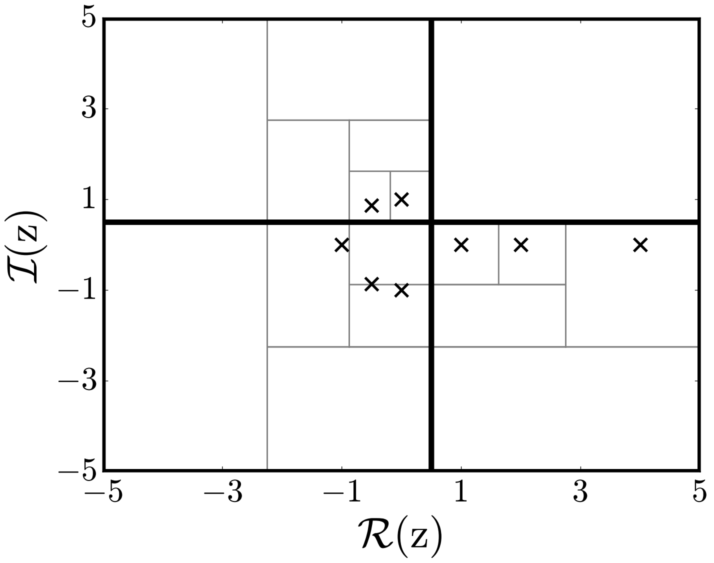

Figure 2.6: Visualisation of the zeros of f_2. Meshes are represented by thin grey lines, except for reshaped ones, which are represented in thick black lines.

2.5.2 Thermoacoustic cases

The two following examples are associated with thermo-acoustic problems formulated by Dellnitz et al [2] and Bauerheim [15].

Example 3:

$$f_3(z) = z^2 + Az + Be^{-Tz} + C \tag{2.5.3}$$

This case concerns the acoustic stability study of a reactive flow in an annular combustion chamber (see [2] for more details), the following set of values is used : $A = -0.19435$, $B = 1000.41$, $C = 522463.0$ and $T = 0.005$. 24 zeros are enclosed in the domain defined by $\mathbf{U} = [Z_{\min} = -5000 - 15000i, Z_{\max} = 5000 + 15000i]$. This computation required 11828 calls to f_3 (AS scheme). This corresponds to approximately 62 ms which can be compared with the 3 seconds obtained in [4] and 83 integrals computations (24 were dedicated to validate the guesses). As a comparison, solving f_3 without the generalized argument principle ($N_{\max} = 0$) required 17588 calls to f_3 and 111 integrals computations.

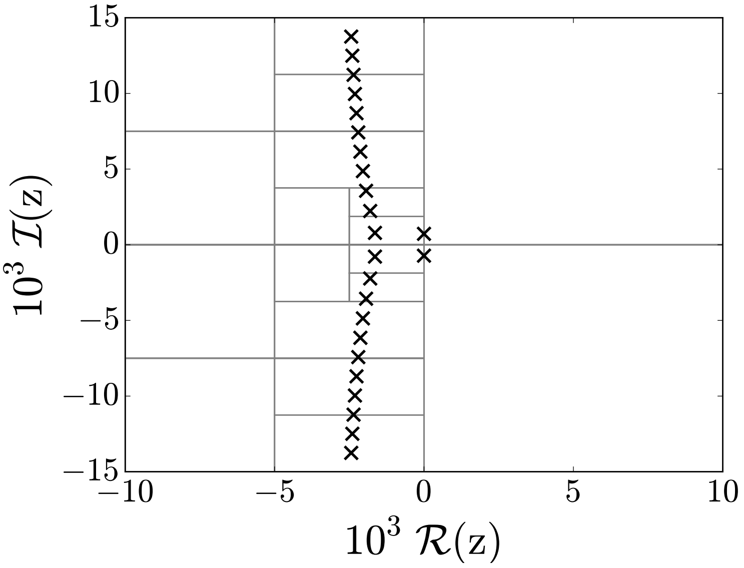

Figure 2.7: Visualisation of the zeros of f_3. Meshes are represented by thin grey lines

Example 4:

The function used in example 4 is proposed by Bauerheim et al [15] and dedicated to predict the thermo-acoustic modes of annular combustion chambers:

$$f_4(z) = \det\left(\prod_{i=4}^{1} \mathrm{R_i}(z)\mathrm{T_i}(z) - \mathrm{Id}\right) \tag{2.5.4}$$

where $\mathrm{R_i}$ designates a four by four rotation matrix and $\mathrm{T_i}$, a four by four transfer matrix which accounts for the flame/acoustic coupling. This model enlightens the ability of RootLocker to deal with realistic characteristic equations. All parameters are described in [15], table 1, except τ, the flame delay, which is set to 5 ms and $n = 1.57$, the interaction index. In symmetrical configurations, when all burners are identical and uniformely distributed around the annular chamber, some acoustic modes are of second multiplicity. Symmetry may be broken by sightly modifying the interaction between the burners and the acoustic [16]. Second order acoustical modes are then split into separate modes. However, the roots of equation 1.5.4 may still stay very close to each other if the asymmetry remains small. In such situations, enclosing several roots from a single mesh without splitting it is very useful.

The domain of interest is defined by $\mathbf{U} = [Z_{\min} = 4{-}20i\ ,\ Z_{\max} = 70{+}22i]$ where z $[Hz]$ is the complex valued frequency. In the symmetrical case, one unstable solution is enclosed at $z_1 = 28.17118 + 4.70674i$ and stable solution is found equal to $z_2 = 57.11084 - 0.32660i$ (multiplicity of two). Computational time and number of calls to f_4 are reproduced in table 1.1. The meshes needed to enclose the roots are displayed in Fig. 1.8 for $N_{\max} \in \{0, 1, 2, 3\}$.

$N_{\max}$ Scheme	0	1	2	3
MR	161996 (5.1s)	132121 (4.1s)	19489 (0.59s)	11153 (0.33s)
R	158015 (7.8s)	157459 (6.8s)	667 (0.03s)	387 (0.02s)
AS	8437 (0.42s)	7972 (0.40s)	406 (0.021s)	241 (0.012s)

Table 2.1: Number of calls to f_4 and associated computational time in the symmetrical case.

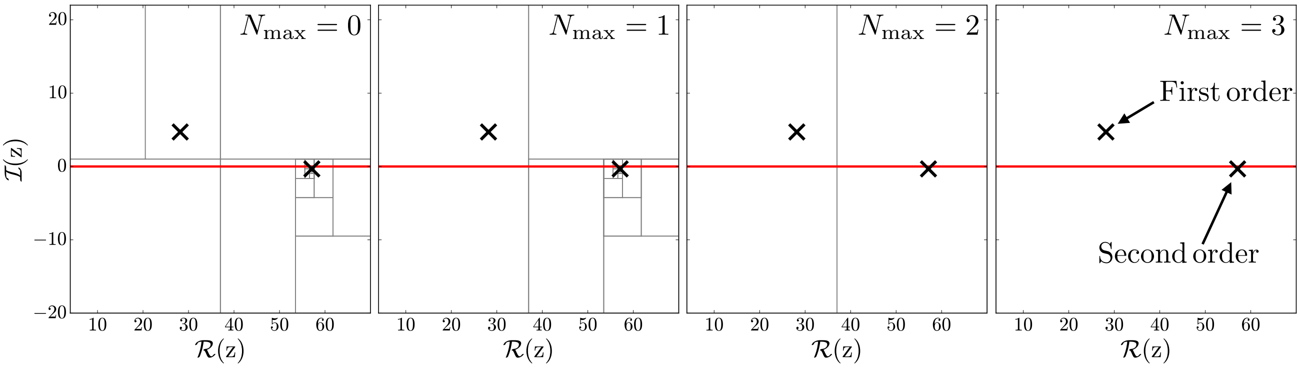

Figure 2.8: Visualisation of the zeros of f_4 in the symmetrical case with the AS scheme. From left to right: $N_{\max} \in \{0, 1, 2, 3\}$.

When $N_{\max} = 0$, contour integrals only provide the number of zeros contained in the meshes. When a mesh contains only one root, it is extracted thanks to the NR algorithm started from the center of the mesh. In the present case, the NR algorithm did not converge and the mesh has been refined two additional times (see Fig. 1.8, $N_{\max} = 0$). When $N_{\max} = 1$, the contour integrals provide a guess to the NR algorithm by solving the one order equivalent polynomial. This guess helps the NR algorithm to converge as the single root was extracted without mesh refinement. However, roots of multiple order are not extracted directly and the concerned meshes are refined until the minimal size is reached. When $N_{\max} \geq 2$, the complete algorithm is applied with equivalent polynomials of order 2 and 3 respectively as shown in Fig. 1.8.

One can notice that the best results are obtained in the case $N_{\max} = 3$ as fewer meshes are needed to enclose all the roots. When $N_{\max} < 2$, using the R scheme takes a longer time than unsing the MR scheme for an equivalent number of calls to f_4 as it implies the computation of derivatives with automated differentiation. However, when $N_{\max} \geq 2$, the R scheme performances reach those of the AS scheme. As calls to f_4 are quite expensive, both R and AS schemes exhibit the same amount of calls per second which is roughly equal to 20000 s^{-1} showing that the overhead caused by the algorithm is negligible.

The asymmetrical case is obtained with the following set of interaction indexes: $\{n_1 = 1.55\ ,\ n_2 = 1.6\ ,\ n_3 = 1.54\ ,\ n_4 = 1.59\}$ so that $\bar{n} = 1.57$, all the other parameters being kept unchanged. The following single roots are enclosed: $\{28.17117 + 4.70662i\ ,\ 57.10970 - 0.33192i\ ,\ 57.11195 - 0.32127i\}$. Computational time and number of calls of f_4 are reproduced in table 1.2.

When $N_{\max} \geq 2$, the numbers of calls to f_4 from both symmetrical and asymmetrical cases are similar. Indeed, enclosing two single solutions instead of a multiple one only implies an additional call to the NR method in this case. The factor 2 between the computational times is justified by the increased complexity of f_4 computations in the asymmetrical case as four different matrices must be assembled instead of one. However, when $N_{\max} < 2$, fewer calls to f_4 are needed to enclose all roots as meshes are not split down to the precision limit as it was the case in the symmetrical case.

The combination AS, $N_{\max} = 3$ always provided the fastest results. For instance, a speed up of

$N_{\max}$ / Scheme	0	1	2	3
MR	132580 (7.1s)	98236 (5.3s)	19673 (1.08s)	11338 (0.69s)
R	75467 (6.7s)	74519 (6.7s)	715 (0.072s)	435 (0.040s)
AS	5392 (0.54s)	4789 (0.51s)	428 (0.045s)	257 (0.026s)

Table 2.2: Number of calls to f_4 and associated computational time in the asymmetrical case.

500 is observed when compared with the worst combination: MR, $N_{\max} = 0$. The presented algorithm can be further improved by increasing $N_{\max}$ up to the number of solutions encountered in the initial domain: $N_{\max} = N$. This can be achieved by solving numerically the equivalent polynomial of order N with an efficient method such as the Jenkins–Traub algorithm [17]. However, this procedure is limited by the accuracy of the polynomial coefficients, determined by contour integrals computations which are bounded to discretization errors.

2.6 Parametric study of a laboratory scale combustion chamber

In this section, we use again the example of the INTRIG ROM derived in chapter 1. After taking exactly the same parameters, 15 single roots are finally enclosed to the precision 10^{-5}. This computation costs 640 (+440 for verifications) calls to the function f_0 defined in equation 1.6.5, which is much less than the number of calls needed for the stochastic A_{NR} algorithm presented in chapter 1 (more than 5000 calls for the same precision). These roots are reproduced in Fig. 1.9 and are identical to the ones obtained in chapter 1.

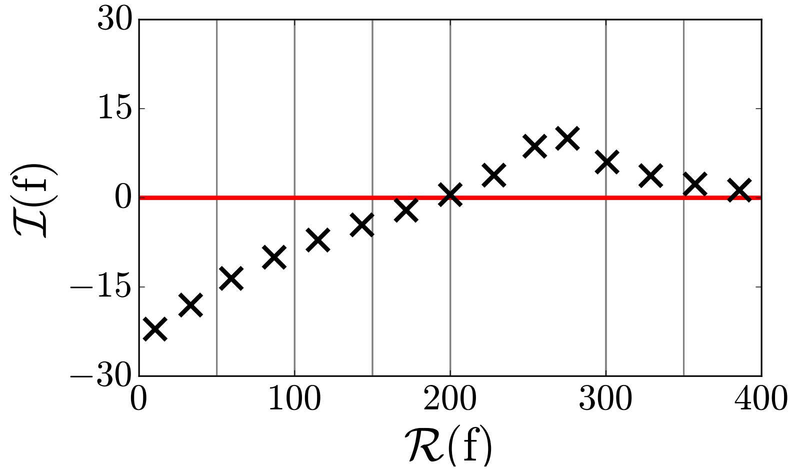

Figure 2.9: Solutions for the INTRIG ROM with un-realistic FTF. The meshes are denoted by thin gray lines

It is now interesting to focus on parametric studies, as they provide information on the acoustic modes of the INTRIG setup. Two kinds of parametric studies can be performed on top of the rootlocker algorithm:

parametric studies

- **Safe parametric mode**:
  In this mode, the A_{RL} algorithm is called for each value of the varied parameter. This ensures that all the solutions are enclosed properly but the cost is proportional to the number of values taken by the parameter.

- **Fast parametric mode**:
  In this mode, the A_{RL} algorithm is only called for the first value of the parameter. The NR method is then called for each successive value, by taking the previous results as initial guesses. This has the major advantage of reducing the cost of the parametric study as all integral computations are removed. However, this method may first fail to capture correctly a mode which crosses an other on the complex plane. Second, it cannot detect new solutions that can appear at the boundary of the domain of research.

In the case of the INTRIG ROM, the A_{RL} algorithm performs very fast so that the safe parametric mode is preferred. The first parametric study concerns the flame delay τ which had been arbitrarily set to 35 ms in chapter 1. This parameter often controls the onset of thermo-acoustic instabilities [18]. It is now varied from 0, which corresponds to a one-dimensional flame approximation up to 50 ms. This study can be performed experimentally by modifying both the equivalence ratio, to variate to flame delay, and the bulk velocity, to maintain the flame shape.

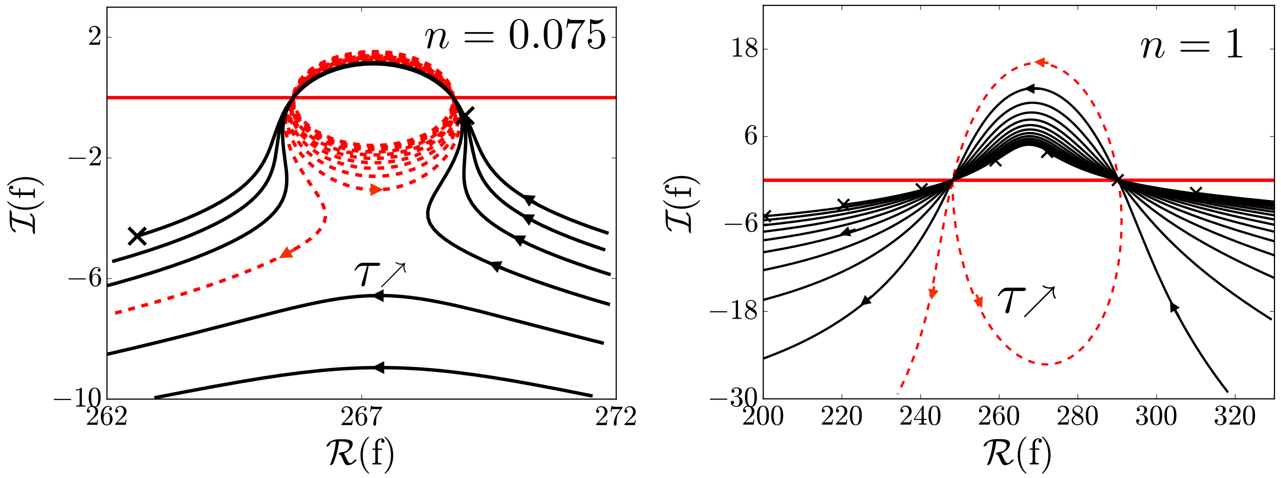

Figure 2.10: Parametric study for the flame delay τ. Studies have been performed with two different interaction index: left $n = 0.075$; right $n = 1$.

In order to study the influence of this parameter, two values for the interaction index are used: $n = 0.075$ and $n = 1$[3]. With the small interaction index, two different kinds of modes are observed. First, there is an acoustic mode which is present even without flame delay (semi-line in Fig. 1.10, left). This mode switches between stable and unstable regimes until $\tau = 40$ ms, when it finally remains stable. This results is coherent with the theoretical stability analysis which can be performed in the low interaction index limit[19]. The other family (thick lines in Fig. 1.10, left) is constituted by the modes which do not exist without flame delay ($\tau = 0$). Some of these modes remain stable, whatever the value of the flame delay. Others are coupled with the first family mode: they exhibit unstable behaviour.

With the interaction index set to unity, the two families of acoustic modes are still observed, but the coupling between them is stronger: the first mode (semi-line in Fig. 1.10, right) stops alternating

[3] Even though values of n larger than unity are sometimes observed, $n = 1$ is a reasonable upper value for parametric studies.

between stable and unstable regimes after only one cycle. For important interaction indices ($n \geq 1$), the stability of the system is not affected by the flame delay anymore: there is always at least one unstable mode close to $f = 270\ Hz$.

The second parametric study concerns the interaction index of the $n - \tau$ model: n, which is varied from zero to unity. The corresponding complex frequencies are displayed in Fig. 1.11. One may notice that only one mode is present when the coupling between the flame and the acoustic waves is removed ($n = 0$). As the interaction index increases, other modes, which are initially under-damped, enter the domain of research and some become unstable. A physical discussion concerning this classification, based on block diagram analysis, is provided in chapter 9.

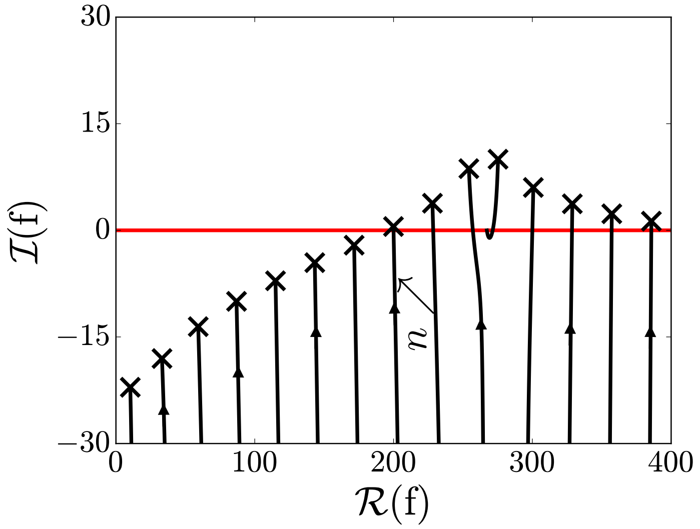

Figure 2.11: Parametric study for the interaction indices $n \in [0, 1]$ in the INTRIG ROM. The lines show the trajectories of the modes when n goes from 0 to 1.

Conclusion

The A_{RL} algorithm dedicated to find all the solutions of a ROM has been detailed and validated. Its Java implementation, called RootLocker, is more efficient than previous solvers based on the principle of the argument. All roots, multiple or distinct, are enclosed with the accuracy prescribed by the user. Among all the variants of the algorithm, the AS scheme of integration, based on the adaptive Simpson scheme, combined with third order ($N_{\max} = 3$) equivalent polynomial resolution, gives the best results in term of computational time (below 0.1 s in all examples) and calls to the function f. This short computational time makes it possible to perform fast but still safe parametric studies. Furthermore, the A_{RL} algorithm used here for the resolution of thermo-acoustic equations is also suitable for other kind of problems implying the resolution of holomorphic equations. Indeed, this method is particularly adapted to solve low-dimension non linear eigen-value problems, or characteristic equations obtained in wave dynamic studies such as hydrodynamics, optics or structure mechanics.

All the resolutions of ROM presented in the following chapters of the manuscript were carried out with the RootLocker tool, which is built upon an intuitive graphical user interface. For instance, it is used in chapter 5, in order to find the acoustic modes of a cavity containing a diaphragm in the presence of a mean flow. The simple ROM of the INTRIG burner presented here as an example permitted to retrieve unstable modes closed to 270 Hz which were observed in the experiments. To go further and provide more quantitative results, a more realistic ROM of the INTRIG burner will be introduced and solved in chapter 8.

Bibliography

[1] L. M. Delves, J. N. Lyness, A numerical method for locating the zeros of an analytic function, Mathematics of Computation 21 (100) (1967) 543–560.

[2] M. Dellnitz, O. Schütze, Q. Zheng, Locating all the zeros of an analytic function in one complex variable, Journal of Computational and Applied Mathematics 138 (2) (2002) 325–333.

[3] T. Johnson, W. Tucker, Enclosing all zeros of an analytic function — A rigorous approach, Journal of Computational and Applied Mathematics 228 (1) (2009) 418–423.

[4] T. Johnson, W. Tucker, On a fast and accurate method to enclose all zeros of an analytic function on a triangulated domain, Proceedings of PARA-2008, to appear in Lecture Notes in Computer Science 6126 (2009) 6127.

[5] J. N. Lyness, Numerical algorithms based on the theory of complex variable, Proceedings of the 1967 22nd national conference on - (1967) 125–133.

[6] T. Suzuki, T. Suzuki, A modification of the numerical integration error method for the zero-finding problem of an analytic function, Japan Journal of Industrial and Applied Mathematics 22 (3) (2005) 353–365.

[7] K. H. Ko, T. Sakkalis, N. M. Patrikalakis, A reliable algorithm for computing the topological degree of a mapping in R2, Applied Mathematics and Computation 196 (2008) 666–678.

[8] J. Kopitz, W. Polifke, CFD-based application of the Nyquist criterion to thermo-acoustic instabilities, Journal of Computational Physics 227 (14) (2008) 6754–6778.

[9] V. Semenov, Method for the Calculation of All Non-Multiple Zeros of an Analytic Function, Comp. Methods in Applied Mathematics 11 (1) (2011) 67–74.

[10] D. Kalman, A matrix proof of newton's identities, Mathematics Magazine 73 (4) (2000) pp. 313–315.

[11] F. Jedrzejewski, Introduction aux méthodes numériques, Springer Science & Business Media, 2006.

[12] J. N. Lyness, Notes on the adaptive simpson quadrature routine, J. ACM 16 (3) (1969) 483–495.

[13] T. Parr, The Definitive ANTLR Reference: Building Domain-Specific Languages., The Pragmatic Programmers, 2007.

[14] M. Bartholomew-biggs, S. Brown, B. Christianson, L. Dixon, Automatic differentiation of algorithms, Journal of Computational and Applied Mathematics 124 (2000) 171–190.

[15] M. Bauerheim, J.-F. Parmentier, P. Salas, F. Nicoud, T. Poinsot, An analytical model for azimuthal thermoacoustic modes in an annular chamber fed by an annular plenum, Combust. Flame 161 (5) (2014) 1374 – 1389.

[16] M. Bauerheim, M. Cazalens, T. Poinsot, A theoretical study of mean azimuthal flow and asymmetry effects on thermo-acoustic modes in annular combustors, Proc. Combust. Inst. 35 (3) (2015) 3219 – 3227.

[17] M. A. Jenkins, J. F. Traub, A three-stage variable-shift iteration for polynomial zeros and its relation to generalized rayleigh iteration, Numerische Mathematik 14 (3) (1970) 252–263.

[18] D. Durox, T. Schuller, S. Candel, Combustion dynamics of inverted conical flames, Proc. Combust. Inst. 30 (2) (2005) 1717 – 1724.

[19] T. Poinsot, D. Veynante, Theoretical and Numerical Combustion, Third Edition (www.cerfacs.fr/elearning), 2011.