Solution of a nonlinear eigenvalue problem from photonic crystal fiber applications discretized by a boundary element method

Ronan Perrussel, Jean-René Poirier

Solution of a nonlinear eigenvalue problem from photonic crystal fiber applications discretized by a boundary element method

Ronan Perrussel[a], Jean-René Poirier[a]

[a]*LAPLACE, University of Toulouse, CNRS, INPT, UPS, 2 rue Camichel, Toulouse, 31071, France*

Abstract

Several strategies for solving a nonlinear eigenvalue problem are evaluated. This problem stems from the boundary integral equation solution of propagation in photonic crystal fibers. The origin and specificities of the eigenvalue problem are recalled before considering the solution of this eigenvalue problem. The first strategy, which is the starting point to illustrate the difficulties, is to solve the problem using Muller's method. We then look at more recent techniques based on contour integrals or a rational interpolant that can be used to compute several eigenmodes simultaneously and considerably reduce the volume of computations.

Keywords: Boundary Element Method, Photonic Crystal Fiber, Nonlinear eigenvalue problem

1. Introduction

Optical fibers and waveguides are important building blocks for many photonic devices and systems in the fields of telecommunications, data transfer and processing, and optical computing. They are widely used as basic components in integrated optical circuits and optical communication systems. In recent years, many complex optical waveguides have emerged, such as photonic crystal fibers [1] or plasmonic waveguides [2]. To reduce the cost of designing new photonic devices, accurate and efficient simulation tools are in great demand in the integrated photonics industry. The first step in photonics simulation is to calculate a complete set of propagation modes accurately and efficiently.

The geometry of these optical waveguides, as well as the dielectric characteristics of the materials and the wavelength of the light source traveling in the fiber, are essential parameters for the propagation of information. These parameters define a complex number called the effective refractive index that characterizes the propagation in the fiber. These new optical fibers have demonstrated an extremely wide range of potential applications in various fields, from defense to environmental applications. The complexity of these systems – heterogeneous structure, waveguide cross-section geometry, and micrometer order of magnitude – makes the numerical methods indispensable to quickly design a PCF for the desired application. The most common methods for solving this type of problems (finite difference time-domain [3], finite element method [4, 5]) can require a huge amount of memory and computation time; these are strong limitations. From 20 years, tools based on the Boundary Element Method (BEM) have been studied for the computation of PCF models.

BEM has a number of advantages: it allows to consider only the mesh at the interfaces of the structure, whereas FEM or finite differences lead to volume discretization and require an artificial boundary condition to take into account the unbounded domain. Despite the fact that the matrix is full (which can now be improved by using well-known acceleration methods such as FMM or H-Matrix), it allows to work with a fine discretization and leads to very accurate solutions.

The BEM for integral equations in (E_z, H_z) has been first considered for PCF in 2004 [6, 7]. This work was improved in [8, 9] with integral equations in (H_x, H_y). More recently a second kind integral equation formulation [10] have been proposed. In this paper we focus on the Nonlinear Eigenvalue Problem (NEP) that has to be solved in each case. In terms of computation time, this is one of the main difficulties of the problem. The cited publications use similar methods to solve the NEP: this involves reducing the matrix problem to a scalar problem and using classical methods for solving nonlinear problems (Newton, Secant or Muller).

In this paper we propose an alternate solver introduced for other integral equations linked to different application domain like acoustics; see for instance [11, 12] for similar approaches to acoustic problems. This solver is based on a contour integral method that has many advantages in comparison to the more conventional solvers cited previously. A less knowledge of the solution is needed (no close starting point) and the computation of several solutions with easy parallelization can be done. In addition no computation

with poor conditionned matrix close to singular one are needed. We present in the first section the boundary-value problem which under consideration and the integral equations used in this work. In section 2, we first study the limits of the method that is proposed in most of the papers using integral equations to deal with PCFs. Section 3 is then devoted to the implementation of a solution method based on a contour integral [13] and its numerical implementation. In the following section an approach with a similar algorithm but based on a rational interpolant of the matrix-valued problem is compared to the previous methods. Finally in the last section we present some numerical tests on particular cases before concluding.

2. Boundary Integral Equations

2.1. Maxwell's Equations and transmission conditions

We consider an optical guide invariant along the direction z with C inclusions $(\Omega_j)_{j=1,\dots,C}$ of refractive index n_j, contained in a material Ω_0 of refractive index n_0. Normal and tangent vectors to inclusion boundaries Γ_j are denoted $\boldsymbol{\nu}$ and $\boldsymbol{\tau}$ (see Figure 1).

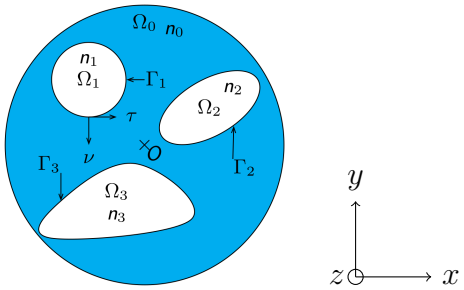

Figure 1: PCF geometry and notations

Following [8] the dependence of the electromagnetic field in z is taken into account by a term $\exp(\mathrm{j}\beta z)$ meaning an $\exp(-\mathrm{j}\omega t)$ dependence with j the imaginary unit, β a propagation constant and ω the angular frequency.

The computation of the PCF eigenmodes leads to satisfy the following Helmholtz equation for each component of the electromagnetic field:

$$\partial_x^2 u + \partial_y^2 u + k^2 u = 0$$

where $k^2 = k_0^2 n_j^2 - \beta^2$ and with $k_0 = 2\pi/\lambda_0$ the wavenumber in free space.

Moreover, the following transmission conditions are satisfied for each Ω_j on the interface Γ_j

$$[\partial_\nu H_\nu]_{\Gamma_j} = [E_z]_{\Gamma_j} = 0, \quad \forall j \in [\![1; C]\!] \tag{1}$$

where $[\cdot]_{\Gamma_j}$ represents the jump through Γ_j.

Taking invariance into account and expressing in terms of H_x and H_y the equation $\nabla \times H = \mathrm{j}\omega\varepsilon E$, we obtain

$$\mathrm{j}k_0 n_j E_z = \partial_y H_x - \partial_x H_y, \quad \forall j \in [\![1; C]\!]. \tag{2}$$

We will also introduce the quantities $k_j = k_0 n_j$ the wavenumber in Ω_j and $n_{\text{eff}} = k_0/\beta$ the effective refractive index.

2.2. Integral Formulation

In this paper, we work with the formulation in (H_x, H_y) described in [8] which is known to be smoother [9] but methods described here are also efficient with other formulations.

To introduce this, we use the free space Green function

$$G(\mathbf{r}, \tilde{\mathbf{r}}) = \frac{\mathrm{j}}{4} H_0^{(1)} \left((k^2 - \beta^2) |\mathbf{r} - \tilde{\mathbf{r}}| \right), \qquad \mathbf{r} \neq \tilde{\mathbf{r}} \tag{3}$$

where $H_0^{(1)}$ is the Hankel function.

Fields $u = H_x$ or H_y are solutions of the Helmholtz equation and can be represented for $\mathbf{r} \notin \partial\Omega_j$ by

$$u(\mathbf{r}) = \pm \left(\int_{\partial\Omega_j} G(\mathbf{r}, \tilde{\mathbf{r}}) \frac{\partial u(\tilde{\mathbf{r}})}{\partial \boldsymbol{\nu}(\tilde{\mathbf{r}})} \mathrm{d}s(\tilde{\mathbf{r}}) - \int_{\partial\Omega_j} \frac{\partial G(\mathbf{r}, \tilde{\mathbf{r}})}{\partial \boldsymbol{\nu}(\tilde{\mathbf{r}})} u(\tilde{\mathbf{r}}) \mathrm{d}s(\tilde{\mathbf{r}}) \right), \tag{4}$$

where $\pm$ becomes $+$ for the interior domain *i.e.* Ω_j and $-$ for the exterior domain *i.e.* Ω_0. It provides, when $\mathbf{r}$ tends to $\partial\Omega_j$, the integral equations

$$\frac{\partial u(\mathbf{r})}{\partial \boldsymbol{\nu}} = \frac{1}{2} \frac{\partial u(\mathbf{r})}{\partial \boldsymbol{\nu}} \pm \left(\int_{\partial\Omega_j} \frac{\partial G(\mathbf{r}, \tilde{\mathbf{r}})}{\partial \boldsymbol{\nu}(\mathbf{r})} \frac{\partial u(\tilde{\mathbf{r}})}{\partial \boldsymbol{\nu}} \mathrm{d}s(\tilde{\mathbf{r}}) - \int_{\partial\Omega_j} \frac{\partial^2 G(\mathbf{r}, \tilde{\mathbf{r}})}{\partial \boldsymbol{\nu}(\tilde{\mathbf{r}}) \partial \boldsymbol{\nu}(\mathbf{r})} u(\tilde{\mathbf{r}}) \mathrm{d}s(\tilde{\mathbf{r}}) \right), \ \mathbf{r} \in \partial\Omega_j. \tag{5}$$

To express the transmission conditions $[\partial_\nu H_\nu]_{\Gamma_j} = [E_z]_{\Gamma_j} = 0$, we define the following operators for each domain Ω_j, $j \in [\![1; C]\!]$, $J_i^{(j)}$ and $W_i^{(j)}$ in the "interior" of the domain

$$\begin{cases} (J_i^{(j)}\psi)(\mathbf{r}) = \displaystyle\int_{\partial\Omega_j} \frac{\partial G(\mathbf{r},\tilde{\mathbf{r}})}{\partial\nu(\mathbf{r})}\psi(\tilde{\mathbf{r}})\,\mathrm{d}s(\tilde{\mathbf{r}}), \quad \mathbf{r} \in \partial\Omega_j, \\ (W_i^{(j)}\psi)(\mathbf{r}) = \displaystyle\int_{\partial\Omega_j} \frac{\partial^2 G(\mathbf{r},\tilde{\mathbf{r}})}{\partial\nu(\mathbf{r})\partial\nu(\tilde{\mathbf{r}})}\psi(\tilde{\mathbf{r}})\,\mathrm{d}s(\tilde{\mathbf{r}}), \quad \mathbf{r} \in \partial\Omega_j. \end{cases} \tag{6}$$

Then, for $\mathbf{r} \in \partial\Omega_j$, $(J_i\psi)(\mathbf{r}) = (J_i^{(j)}\psi)(\mathbf{r})$ and $(W_i\psi)(\mathbf{r}) = (W_i^{(j)}\psi)(\mathbf{r})$.

J_e and W_e in the "exterior" of the domains are defined by

$$\begin{cases} (J_e\psi)(\mathbf{r}) = \displaystyle\sum_{l=1}^{C} \int_{\partial\Omega_l} \frac{\partial G(\mathbf{r},\tilde{\mathbf{r}})}{\partial\nu(\mathbf{r})}\psi(\tilde{\mathbf{r}})\,\mathrm{d}s(\tilde{\mathbf{r}}), \quad \mathbf{r} \in \displaystyle\bigcup_{j=1}^{C} \partial\Omega_j, \\ (W_e\psi)(\mathbf{r}) = \displaystyle\sum_{l=1}^{C} \int_{\partial\Omega_l} \frac{\partial^2 G(\mathbf{r},\tilde{\mathbf{r}})}{\partial\nu(\mathbf{r})\partial\nu(\tilde{\mathbf{r}})}\psi(\tilde{\mathbf{r}})\,\mathrm{d}s(\tilde{\mathbf{r}}), \quad \mathbf{r} \in \displaystyle\bigcup_{j=1}^{C} \partial\Omega_j. \end{cases} \tag{7}$$

Note that in the exterior of the domains Ω_j, for all $j \in [\![1; C]\!]$, – *i.e.* in the medium Ω_0 – each domain interacts with all the others when for the interior of the domain Ω_j it interacts only with itself (leading to a block-diagonal matrix after discretization).

For simplicity we note

$$\begin{cases} H_x = \mu^{H_x} \quad \text{and} \quad \dfrac{\partial H_x}{\partial \boldsymbol{\nu}_i}(\mathbf{r}) = \sigma_i^{H_x}, \quad \dfrac{\partial H_x}{\partial \boldsymbol{\nu}_e}(\mathbf{r}) = \sigma_e^{H_x} \\ H_y = \mu^{H_y} \quad \text{and} \quad \dfrac{\partial H_y}{\partial \boldsymbol{\nu}_i}(\mathbf{r}) = \sigma_i^{H_y}, \quad \dfrac{\partial H_y}{\partial \boldsymbol{\nu}_e}(\mathbf{r}) = \sigma_e^{H_y} \end{cases} \tag{8}$$

which leads to

$$\begin{cases} N_i\sigma_i^{H_k} = -W_i\mu^{H_k} \\ N_e\sigma_e^{H_k} = W_e\mu^{H_k} \end{cases}, \quad \text{for } k \in \{x, y\}, \tag{9}$$

with $N_i = \dfrac{1}{2}Id - J_i$ and $N_e = \dfrac{1}{2}Id + J_e$ and Id is the identity operator. Using $\partial_\nu H_\nu = \nu_x\partial_\nu H_x + \nu_y\partial_\nu H_y$, we express the boundary condition by

$$[\partial_\nu H_\nu] = \nu_x\,[\partial_\nu H_x] + \nu_y\,[\partial_\nu H_y] \Rightarrow \nu_x(\sigma_i^{H_x} - \sigma_e^{H_x}) + \nu_y(\sigma_i^{H_y} - \sigma_e^{H_y}) = 0 \tag{10}$$

that can be put under the form

$$\nu_x \left(N_i^{-1}W_i + N_e^{-1}W_e\right) \mu^{H_x} + \nu_y \left(N_i^{-1}W_i + N_e^{-1}W_e\right) \mu^{H_y} = 0. \tag{11}$$

Then, with $\partial_y = \nu_y \partial_\nu + \nu_x \partial_\tau$ and $\partial_x = \nu_x \partial_\nu - \nu_y \partial_\tau$, equation (2) becomes

$$\mathrm{j}k_0 n^2 E_z = \left(\nu_y \partial_\nu + \nu_x \partial_\tau\right) H_x + \left(\nu_x \partial_\nu - \nu_y \partial_\tau\right) H_y \tag{12}$$

and we get

$$\begin{cases} \mathrm{j}k_0 n_i^2 E_z = (\nu_y \sigma_i^{H_x} + \nu_x \partial_\tau \mu^{H_x} + \nu_x \sigma_i^{H_y} - \nu_y \partial_\tau \mu^{H_y}) \\ \mathrm{j}k_0 n_e^2 E_z = (\nu_y \sigma_e^{H_x} + \nu_x \partial_\tau \mu^{H_x} + \nu_x \sigma_e^{H_y} - \nu_y \partial_\tau \mu^{H_y}) \end{cases}. \tag{13}$$

The last boundary condition $[E_z] = 0$ gives

$$\begin{aligned} \nu_y \left(\frac{\sigma_i^{H_x}}{n_i^2} - \frac{\sigma_e^{H_x}}{n_e^2}\right) + \nu_x \left(\frac{1}{n_i^2} - \frac{1}{n_e^2}\right) \partial_\tau \mu^{H_x} \\ + \nu_x \left(\frac{\sigma_i^{H_y}}{n_i^2} - \frac{\sigma_e^{H_y}}{n_e^2}\right) - \nu_y \left(\frac{1}{n_i^2} - \frac{1}{n_e^2}\right) \partial_\tau \mu^{H_y} = 0 \end{aligned} \tag{14}$$

with $B_1 = N_i^{-1}W_i$ and $B_2 = N_e^{-1}W_e$, the condition becomes

$$\begin{aligned} -\nu_y \left(\frac{B_1}{n_i^2} + \frac{B_2}{n_e^2}\right) \mu^{H_x} + \nu_x \left(\frac{1}{n_i^2} - \frac{1}{n_e^2}\right) \partial_\tau \mu^{H_x} \\ - \nu_x \left(\frac{B_1}{n_i^2} + \frac{B_2}{n_e^2}\right) \mu^{H_y} - \nu_y \left(\frac{1}{n_i^2} - \frac{1}{n_e^2}\right) \partial_\tau \mu^{H_y} = 0 \end{aligned} \tag{15}$$

Maintaining similar notations, we now consider that the continuous equations introduced above are discretized by a boundary element approach, with operators and fields becoming matrices and vectors. In this discretization, we point out that the computation of the B_2 matrix implies a full matrix solution of half the size of the final matrix, which has a non-negligible cost during assembly. Using the discretized tangential operator $T \sim \partial_\tau$ we get

$$\begin{bmatrix} A_f & B_f \\ C_f & D_f \end{bmatrix} \begin{bmatrix} \mu^{H_x} \\ \mu^{H_y} \end{bmatrix} = 0, \tag{16}$$

with

$$\begin{cases} A_f = \nu_x(B_1 + B_2) \\ B_f = \nu_y(B_1 + B_2) \\ C_f = \left(\frac{1}{n_i^2} - \frac{1}{n_e^2}\right)\nu_x T - \nu_y\left(\frac{B_1}{n_i^2} + \frac{B_2}{n_e^2}\right) \\ D_f = -\left(\frac{1}{n_i^2} - \frac{1}{n_e^2}\right)\nu_y T - \nu_x\left(\frac{B_1}{n_i^2} + \frac{B_2}{n_e^2}\right) \end{cases}$$

which leads to the NEP we are interested in.

2.3. The nonlinear eigenvalue problem

Accounting the fact that A_f, B_f, C_f, and D_f depend non linearly on n_{eff} (in particular see (3) where β is inversely proportional to n_{eff}), the final form of the system after discretization is

$$F(n_{\text{eff}})\boldsymbol{\mu} = 0, \tag{17}$$

with

$$F(n_{\text{eff}}) = \begin{bmatrix} A_f & B_f \\ C_f & D_f \end{bmatrix} \in \mathbb{R}^{N_t \times N_t}.$$

This is a Nonlinear Eigenvalue Problem (NEP). A mode of the PCF is a couple $(n_{\text{eff}}, \boldsymbol{\mu})$ that makes the matrix F singular. The problem consists in solving the non-linearity depending on n_{eff} and then to compute the associate subspace of the kernel to determine the field.

2.3.1. A solution with Muller's algorithm

To solve this problem, most of the reference papers [6, 7, 8, 9, 10] propose a method that therefore consists in finding the zeros of the fonction $f(n_{\text{eff}})$ determined by 2 complex random vectors $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ such that.

$$f(n_{\text{eff}}) = \frac{1}{\boldsymbol{\phi}^T F^{-1}(n_{\text{eff}})\boldsymbol{\psi}} \text{with } \boldsymbol{\phi}, \boldsymbol{\psi} \in \mathbb{R}^{N_t}, F(n_{\text{eff}}) \in \mathbb{R}^{N_t \times N_t}.$$

Note that in terms of computation time the term $F^{-1}(n_{\text{eff}})\boldsymbol{\psi}$ will lead to solve a linear system. Solving this linear system may be accelerated by tools like hierarchical matrices [15] or the fast multipole method.

To solve the nonlinear equation $f(n_{\text{eff}}) = 0$, the well-known Muller's algorithm [16] is almost always considered. This method being only valuable

for local minima, some strategies (like a mapping of the parameter) can be considered to start the research close to the solution [17].

In order to evaluate and compare with the methods proposed in the following section, we make in this section a numerical study (that was not available in previous works) of the efficiency of Muller's algorithm.

2.3.2. Some numerical results with Muller's algorithm

The numerical tests in this section correspond to several discretizations of the fiber Figure 2.

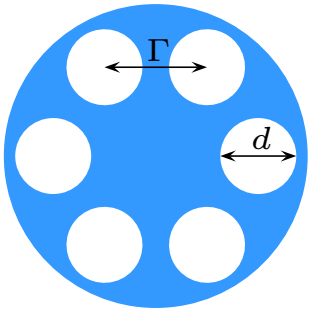

Figure 2: PCF hexapole for numerical tests, $d = 5\mu m$ and $\Gamma = 6.75\mu m$.

The frequency is such that the free space wavelength is $\lambda_0 = 1.45\mu m$. The refractive index of the glass matrix n_e is 1.45 and the medium surrounding the hole is assumed to be infinite.

Table 1 gives the two solutions and the accuracy obtained with this method. It is in good agreement with [8]. The discretization parameter N is the number of unknowns in one hole (the corresponding size of the matrix N_t being then in this case $12N$).

Note that some applications would typically correspond to a refractive index variation of 10^{-6} to 10^{-7} in refractive index units (RIUs) in the external sensitized region of the photonic crystal fiber, hence the specification for a 8-decimal RI resolution. In the following we will keep $N = 80$ for our experiments but less accuracy can be enough.

Let us now study the quality of the error criterion. In order to define a criterion that do not depend on the random vector, we will also characterize the error by the value $|\sigma_{\min}|$ of the smallest singular value of the matrix F. To reduce the influence of the choice of the random vectors $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ it may

N	$n_{\text{eff}\,1}$	$n_{\text{eff}\,2}$
20	$1.44539523115161 + 3.177733\ 10^{-8}$ j	$1.44539523194428 + 3.177861\ 10^{-8}$ j
40	$1.44539523155976 + 3.194522\ 10^{-8}$ j	$1.44539523154495 + 3.194524\ 10^{-8}$ j
80	$1.44539523155975 + 3.194521\ 10^{-8}$ j	$1.44539523154494 + 3.194523\ 10^{-8}$ j
160	$1.44539523155975 + 3.194521\ 10^{-8}$ j	$1.44539523154494 + 3.194523\ 10^{-8}$ j

Table 1: Accuracy of the discretization

be convenient to take several vectors (10 or 20) and compute

$$f_m(n_{\text{eff}}) = \frac{1}{m \left\| \mathbf{\Phi}^T F^{-1}(n_{\text{neff}}) \mathbf{\Psi} \right\|_2} \text{ with } \mathbf{\Phi}, \mathbf{\Psi} \in \mathbb{R}^{m \times N_t}.$$

Table 2 provides the values of the criteria relatively to n_{eff} (taken with some significant figures).

$f_{10}(n_{\text{eff}})$	$f(n_{\text{eff}})$	$\lvert\sigma_{\min}\rvert$	$\lvert\sigma_{\max}\rvert$	n_{eff}
295.9	296.85	2636.8	$2.42\ 10^7$	1.445395
56.82	56.96	506.23	$2.42\ 10^7$	1.4453952
3.15	3.155	28.00	$2.42\ 10^7$	$1.44539523 + 3\,10^{-8}$ j
0.6966	0.700	6.17	$2.42\ 10^7$	$1.445395231 + 3.2\,10^{-8}$ j
0.09209	0.0846	0.6824	$2.42\ 10^7$	$1.4453952316 + 3.19\,10^{-8}$ j
0.008918	0.00759	0.05406	$2.42\ 10^7$	$1.44539523156 + 3.195\,10^{-8}$ j
$9.85\ 10^{-4}$	$5.259\ 10^{-4}$	0.00368	$2.42\ 10^7$	$1.445395231560 + 3.1945\,10^{-8}$ j
$1.5\ 10^{-4}$	$8.40\ 10^{-5}$	$5.88\ 10^{-4}$	$2.42\ 10^7$	$1.4453952315597 + 3.19452\,10^{-8}$ j
$4.42\ 10^{-6}$	$2.41\ 10^{-6}$	$1.68\ 10^{-5}$	$2.42\ 10^7$	$1.44539523155975 + 3.194521\,10^{-8}$ j

Table 2: Accuracy of the error criterion

As a result, the function f seems to be a good tool for finding the roots or for evaluating the accuracy without computing the Singular Value Decomposition (SVD).

We will now illustrate the following three limits of this approach

- Starting point has to be "very" close of a solution to converge;
- Finding multiple zeros is not easy;
- A large number of iterations is needed.

The result of this procedure is random and depends strongly on the starting point. To evaluate (roughly) its influence, we run a number of simulations (50) and then compute the mean number of iterations ($\text{niter}_{\text{mean}}$) and the percentage of runs (%) where the algorithm reach solution 1, solution 2 (previously defined) or another local minimum. We also note when it does not converge.

Table 3, 4 and 5 gives respectively the results of the approach relatively to the number of exact digits for the initial value for Muller's algorithm.

	%	$\text{niter}_{\text{mean}}$
Sol 1	2	111
Sol 2	4	97
Others	20	
no convergence	74	

Table 3: Results of Muller's algorithm with an initial value of 5 significant digits: 1.4453

	%	$\text{niter}_{\text{mean}}$
Sol 1	40	105
Sol 2	35	87.6
Others	25	

Table 4: Results of Muller's algorithm with an initial value of 6 significant digits: 1.44539

In the case with an initial vector with 5 significant digits, the majority of the simulations does not converge. With one supplementary significant digit for the initial vector, 25 % of the simulations do not still converge to one of the solutions. To obtain a convergence for each simulation, it is necessary to start from 1.445395 as shown in Table 5.

	%	$\text{niter}_{\text{mean}}$	$\text{niter}_{\text{min}}$	$\text{niter}_{\text{max}}$
Sol 1	45	92	75	117
Sol 2	55	94.36	68	121
Others	0			

Table 5: Results of Muller's algorithm with an initial value of 7 significant digits: 1.445395

Note that the convergence can be slightly improved with the use of several random vectors as shown in Table 6 but it does not allow to start far from the solution.

Significant digits	$\text{niter}_{\text{mean}}$ for f	$\text{niter}_{\text{mean}}$ for f_{10}
12	27.3	24.9
10	58.1	48.1
8	76.5	71.8

Table 6: Number of iterations relatively to the initial vector.

Note also that we have no control over the number of roots we can find, and this can be a problem if the number is large. The goal of the following sections is to introduce other approaches that can strongly overpass some limitations of Muller's method.

3. Contour integral methods for nonlinear eigenvalue problems

3.1. Theoretical background

For solving a nonlinear eigenvalue problem of the form $F(z)\boldsymbol{\mu} = 0$, approaches exploiting contour integrals have been introduced in [13] and some improved variants have been proposed in several references, see for instance [18]. We will not describe here the whole theory but the reader can found it in the referred papers.

These algorithms exploit Keldysh's theorem that states that, in a domain closed by a smooth contour $\mathcal{C}$, $F(z)^{-1}$ can be written as a sum of a rational function whose poles inside $\mathcal{C}$ are the eigenvalues of $F(z)$ and an analytic functions with no pole inside $\mathcal{C}$. For instance in the case of simple eigenvalues, we can write in a neighborhood $\mathcal{U}$ of $\mathcal{C}$

$$F(z)^{-1} = \sum_{i=1}^{n(\mathcal{C})} v_i w_i^H \frac{1}{z-\lambda_i} + \mathcal{N}(z), \quad z \in \mathcal{U}\backslash\{\lambda_1, \cdots, \lambda_{n(\mathcal{C})}\}.$$

where $n(\mathcal{C})$ is the number of eigenvalues inside $\mathcal{C}$, λ_i are these eigenvalues, v_i and w_i the corresponding right and left eigenvectors and $\mathcal{N}$ is an analytic function. Thanks to the residue theorem, computing contour integrals enables to compute for any analytic function f

$$\frac{1}{2\pi \mathrm{j}} \int_{\mathcal{C}} f(z) F(z)^{-1} dz = \sum_{i=1}^{n(\mathcal{C})} f(\lambda_n) v_n w_n^H \, .$$

An appropriate choice of several functions f will enable us to derive a system of equations relating to λ_n, the left-hand side of which is computed by a numerical quadrature.

The algorithms then consider integrals of the form

$$\frac{1}{2\pi \mathrm{j}} \int_{\mathcal{C}} f(z) L^H F(z)^{-1} R \, dz$$

where $\mathcal{C}$ is a smooth contour enclosing the eigenvalues of interest and L and R are left and right probing matrices (with N_R random columns) *i.e.* L and $R \in \mathbb{R}^{N_t \times N_R}$. The use of $L^H F(z)^{-1} R$ instead of $F(z)^{-1}$ will reduce computation time and memory storage of the problem. The choice of N_R is a key parameter of the method that will be studied in the numerical section.

The residue theorem enables to focus only on the contribution of the rational part. Indeed, the following relation holds

$$\frac{1}{2\pi \mathrm{j}} \int_{\mathcal{C}} f(z) L^H F(z)^{-1} R \, dz = L^H V f(\Lambda) W^H R \tag{18}$$

where $\Lambda = \mathrm{diag}(\lambda_1, \cdots, \lambda_{n(\mathcal{C})})$ is a diagonal matrix containing the eigenvalues inside $\mathcal{C}$ (counting multiplicities) and V and W are matrices collecting right and left eigenvectors. In practice, integrals in (18) are evaluated by numerical quadrature and

$$L^H V f(\Lambda) W^H R \approx \sum_{i=1}^{N_Q} \omega_i f(z_i) L^H F(z_i)^{-1} R,$$

where $\{\omega_i\}_{i=1}^{N_Q}$ and $\{z_i\}_{i=1}^{N_Q}$ are respectively N_Q quadrature weights and points.

In most of the proposed algorithms, f is chosen as the moments z^j, and we can define the N_m moments

$$A_j = L^H V \Lambda^j W^H R, \quad \forall j \in [\![0; 2N_m - 1]\!] \tag{19}$$

and the Hankel and the shifted Hankel matrices

$$\mathbb{H} = \begin{pmatrix} A_0 & A_1 & \dots & A_{N_m-1} \\ A_1 & A_1 & \dots & A_{N_m} \\ \vdots & \vdots & \ddots & \vdots \\ A_{N_m-1} & A_{N_m} & \dots & A_{2N_m-2} \end{pmatrix} \text{ and } \mathbb{H}_s = \begin{pmatrix} A_1 & A_2 & \dots & A_{N_m} \\ A_2 & A_3 & \dots & A_{N_m+1} \\ \vdots & \vdots & \ddots & \vdots \\ A_{N_m} & A_{N_m+1} & \dots & A_{2N_m-1} \end{pmatrix}.$$

Thus, the nonlinear eigenvalue problem has been converted in a linear generalized eigenvalue problem for the pencil $\mathbb{H}_s - \lambda\mathbb{H}$.

As an illustration, let us consider moments $j = 0$ and $j = 1$ and L equals to the identity matrix, we have

$$A_0 = VW^H R$$
$$A_1 = V\Lambda W^H R$$

Computing the truncated Singular Value Decomposition (SVD) of $A_0 = V_0\Sigma_0 W_0^H$ allows to reduce A_1 to an equivalent linear eigenvalue problem of reduced size relatively to the matrix

$$B = V_0^H A_1 W\Sigma_0^{-1} = S\Lambda S^{-1}$$

with $S = V_0^H V$.

By storing also the moments

$$M_j = \sum_{i=1}^{N_Q} \omega_i z_i^j F(z_i)^{-1} R,\ \forall j \in [\![0; N_m - 1]\!],$$

we can also recover the corresponding right eigenvectors. It enables in particular to compute the residue $\|F(\lambda_i)v_i\|_2$ where λ_i and v_i are the eigenvalue and right eigenvector obtained by the algorithm.

3.2. Algorithm

The previous theoretical results leads to Algorithm 1.

Note that the truncation rules and threshold in step 8 will determine the number of computed eigenvalues and the number of residuals to compute in step 9.

The main parameters and theirs impacts to the computation time in this algorithm are

- the size of random matrices $N_t \times N_R$ that determines the number of right-hand sides in the linear system;
- the number of quadrature points N_Q that determines the number of linear systems to solve;
- the number of moment N_m that only impacts the sum in step 5 (additional cost is small).

Algorithm 1 Contour Integral Solver

1: Compute quadrature/sampling points $z_i, i = 0, 1, ..., N_Q - 1$ and corresponding weights ω_i;
2: Generate random matrices L and $R \in \mathbb{R}^{N_t \times N_R}$;
3: Compute $F(z_i)^{-1}R$ for $i = 0, 1, ..., N_Q - 1$;
4: Compute moments $A_\alpha = \sum_{i=0}^{N_m-1} \omega_i z_i^\alpha L^H F^{-1}(z_i)R, \quad \alpha = 0, \ldots, 2N_m - 1$;
5: Compute $M_\alpha = \sum_{i=0}^{N_m-1} \omega_i z_i^\alpha F(z_i)^{-1}R \quad \alpha = 0, \ldots, N_m - 1$;
6: Construct two block Hankel matrices $\mathbb{H}$ and $\mathbb{H}_s$;
7: Construct $M = \begin{pmatrix} M_0 & M_1 & \cdots M_{N_m-1} \end{pmatrix}$;
8: Compute the truncated SVD of $\mathbb{H} = V_0 \Sigma_0 W_0^H$
9: Solve the eigenvalue problem for $B = V_0^H \mathbb{H}_\mathbb{S} W_0 \Sigma_0^{-1}$ and compute the residual $\|F(\lambda_j)v_j\|$ for each value to select the correct ones.
10: For each selected eigenpair (b, λ) the corresponding eigenpair of the original NEP is given by $(MW_0\Sigma_0^{-1}b, \lambda)$;

The aim of the following section will be to study the influence of those parameters on computation time and accuracy.

Note that the sampling in step 3 does not need to have values very close to solution. This is a great advantage because the resolution of the linear system is difficult because of a bad condition number. It's not the case for an iterative algorithm like Muller that tends to this in the last steps

3.3. Numerical Results

3.3.1. Random Solution Study

The first parameter we choose to study is N_R because it may also impact on the random solution. An appropriate choice of N_R will then be required to conclude on the influence of the other parameters. We choose a circle of diameter the real segment $[1.44, 1.45]$ with $N_Q = 20$ for the number of quadrature points. Table 7 provides information on the accuracy of the calculation over a set of 10 simulations with several values of N_R. In this table, $\overline{n_{\text{eff}}}$ is the mean over the 10 computed effective indices $(n_{\text{eff}^{(i)}})_{i=1}^{10}$, σ is the standard deviation over the same set of values and E is the mean of the error relative to a reference value $n_{\text{eff}_{\text{ref}}}$,

$$E = \frac{1}{10} \sum_{i=1}^{10} |n_{\text{eff}^{(i)}} - n_{\text{eff}_{\text{ref}}}|.$$

N_R	$\overline{n_{\text{eff}}}$	σ	E
10	$1.445411055710 + 9.0114\ 10^{-5}$ j	$4.08\ 10^{-4}$	$1.02\ 10^{-4}$
20	$1.445398051119 - 1.7929\ 10^{-6}$ j	$1.22\ 10^{-5}$	$5.27\ 10^{-6}$
40	$1.445395233572 + 3.4528\ 10^{-8}$ j	$3.57\ 10^{-8}$	$2.56\ 10^{-8}$
40	$1.445395233572 + 3.4528\ 10^{-8}$ j	$3.57\ 10^{-8}$	$2.56\ 10^{-8}$
80	$1.445395231848 + 3.2321\ 10^{-8}$ j	$1.04\ 10^{-8}$	$6.36\ 10^{-9}$
160	$1.445395231149 + 3.2907\ 10^{-8}$ j	$3.50\ 10^{-9}$	$2.27\ 10^{-9}$

Table 7: Accuracy relatively to N_R

As it was expected, it is clear from this table that increasing the number of random vectors increases accuracy but also reduces the "random part" of the solution (which is characterized by the standard deviation σ). According to this study, we will take $N_R = 40$ or more (i.e. random vectors) in the following. This will impact moderately the computation cost (see next section) because of the number of right-hand sides to solve.

3.3.2. Discretization of the contour

We are now interested in the the number of quadrature points N_Q on the contour, which is still a circle of diameter $[1.44, 1.45]$. As before, the study is performed for a set of 10 simulations and to limit the randomness of the calculation we take N_R fixed at 50. The results are provided in Table 8 where E has the same definition as in Table 7.

N_Q	E
10	$8.33\ 10^{-7}$
20	$6.66\ 10^{-9}$
40	$4.83\ 10^{-9}$
160	$2.13\ 10^{-9}$

Table 8: Accuracy relatively to N_Q

We can observe a good accuracy with a small number of quadrature points with a strong decrease of the value of E up to $N_Q = 20$. This parameter is the most significant for the accuracy of the solution, but it is also the main parameter for computation time, due to the number of linear systems to be solved.

3.3.3. Computation times

Let us now observe the influence of both parameters on computation times. As written previously, the influence becomes from

- the number of linear system to solve when increasing N_Q, see Table 9;
- the number of eigenvalues to select when increasing N_R, see Table 10.

In Tables 9 and 10, "Solver time" denotes the times required to solve the linear systems and "Residuals time" is the time required to compute the residuals which enables to select the correct eigenmodes.

N_Q	Solver time (s)	Residuals time (s)
10	24	18
20	45	16
40	89	20
80	186	22
160	394	23

Table 9: $N_R = 50$

N_R	Solver time (s)	Residuals time (s)	Neigen
10	24	18	8
20	23	17	9
40	23	21	10
80	23	50	23
160	23	220	96

Table 10: $N_C = 20$

The results in Table 9 are as expected and depends linearly on the parameter N_Q. From Table 10 we can observe that increasing the number of right-hand sides seems to have a reduced influence on the solver time. The main observation is that it increases the number of eigenvalues computed by the algorithm and for large values of N_R this has a non negligible influence on the computation times because of the number of residuals to compute for a selection of the correct eigenvalues. Therefore, choosing around $N_R = 40$ seems to be a reasonable compromise for maintaining good accuracy without significantly increasing computation times.

Note that in [14] a procedure to select and reduce the number of singular values is proposed.

3.3.4. Size of the contour (diameter position)

We now study the influence of the contour and namely the diameter and center position on the accuracy in Tables 11 and 12. It can also be seen as a study of the initial condition for the solution. Note that a diameter is the real segment $[a_x, b_x]$. Other parameters remain unchanged and are the results of choices made in previous studies.

$[a_x, b_x]$	$n_{\text{eff}1}$
[1.44,1.45]	1.445406464954 - 2.86876 10^{-5} j
[1.445,1.446]	1.445395231549 + 3.19465 10^{-8} j
[1.4453,1.4454]	1.445395231559 + 3.19445 10^{-8} j
[1.44539,1.44540]	1.445395231559 + 3.19452 10^{-8} j

Table 11: Reducing interval size

$[a_x, b_x]$	n_{eff_2}
[1.44,1.45]	1.445391106380 + 7.11983 10^{-5} j
[1.445,1.446]	1.445395231500 + 3.20273 10^{-8} j
[1.4453,1.4454]	1.4453952315457 + 3.19468 10^{-8} j
[1.44539,1.44540]	1.4453952315441 + 3.19463 10^{-8} j

Table 12: Reducing interval size

As it was expected, a good knowledge of the behaviour of the solution can strongly increase the accuracy. This observation is interesting, as it suggests that it may be appropriate to use the method iteratively (see last section of the paper).

4. Rational Interpolation Approach

4.1. Theoretical principles

Using the properties of the Keldish theorem which links the eigenvalues of $F(z)$ to the poles of $F(z)^{-1}$, another approach is proposed in [14]. In this reference, the authors prefer to consider a pole-finding method using a rational interpolant and apply it to a reduced matrix function having the same pole as $F(z)$ *i.e.*

$$G(z) = L^H F(z)^{-1} R$$

where L and R are random matrix of full rank used to condense $F(z)^{-1}$ and reduce the computation times.

The poles will be computed by Jacobi's method [19] applied to a rational interpolant of $G(z)$ such that

$$\frac{P(z_i)}{Q(z_i)} = G(z_i), \quad i = 0, \cdots, N_Q - 1$$

where $P(z)$ and $Q(z)$ are polynomial functions.

Jacobi's method applied to $G(z)$ will consider moments of the form

$$A_\alpha = \sum_{i=0}^{N_Q-1} \omega_i z_i^\alpha G(z_i), \quad \alpha = 0, \cdots, 2N_m - 1$$

that is similar to (19) but where ω_i denotes the barycentric weights of the interpolation formula and not quadrature weights, *i.e.*

$$\omega_i = \prod_{l=1, l\neq i}^{N_Q} \frac{1}{z_l - z_i}.$$

The proposed modification leads exactly to the same Algorithm 1 but enables now to choose "quadrature" points not only on the contour $\mathcal{C}$ but also inside.

Considering the particular form of the solutions, this approach will allow to simplify the research and consider points only on the real axis because the imaginary part is very small compared to the real part and the range of value considered for the real part.

Indeed in our application the imaginary part represents the losses which are of small amplitude and thus the value 0 is a good approximation on a large research interval for the real part.

On a smaller research interval it would then be wise to take a segment parallel to the real axis positioned at an approximation of the imaginary part. This can be done in a second step if we want to improve the accuracy of the solution.

4.2. Numerical Results

In this subsection we consider the Rational Interpolant Approach (RIA) on a line (see Figure 3 left) which is compared with the previous Contour Integral Approach (CIA) for several contours (see Figure 3 right).

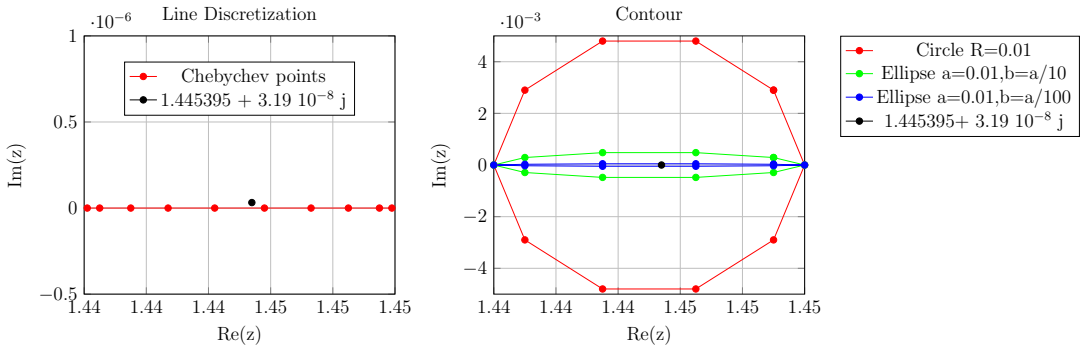

Figure 3: Quadrature or Sampling points on several contours for the RIA (left) and CIA (right)

4.2.1. RIA vs ellipse

We now consider as contour an ellipse with axes a and b to better approximate the imaginary part, which is close to 0, and we compare in Table 13 the performances obtained with RIA.

N_R	$E = \frac{1}{10}\sum_{n=1}^{10} \mid n_{\text{eff}^{(n)}} - n_{\text{eff}_{\text{ref}}} \mid$
Circle	$1.12\ 10^{-8}$
Ellipse $b = \frac{a}{10}$	$1.88\ 10^{-9}$
Ellipse $b = \frac{a}{100}$	$4.55\ 10^{-9}$
RIA	$1.27\ 10^{-11}$

Table 13: Accuracy of RIA vs CIA with several contours

The main observation is that using an ellipse close to the solution does not improve strongly the accuracy when the RIA is obviously better with more significant digits.

4.2.2. holocore Case

At last we apply (iteratively) the method on a larger problem (see Figure 4). The physical parameters of the problem are: $\lambda_0 = 1.51\mu m$, $n_i = 1$, and $n_e = 1.45$.

The BEM is applied with a discretization of 32 points on each hole, except on the central hole where 80 points are considered. It leads to a matrix of size $N_t = 7840$. Note that we do not reach the optimal discretization but it is difficult to increase the number of points without a strong increase of computation times and memory storage. For the nonlinear solver parameters,

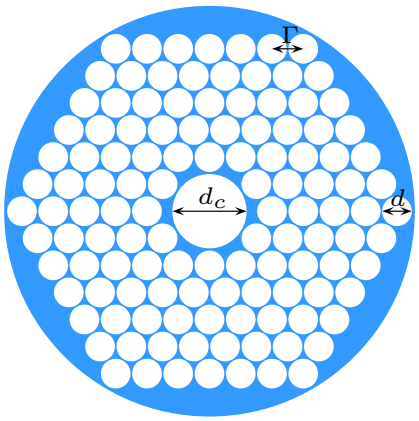

Figure 4: Holocore PCF. $d = 2.603\mu m$, $d_c = 6.5075\mu m$, and $\Gamma = 2.74\mu m$

we take $N_R = 40$, $N_m = 10$ for this simulation. Results from CIA and RIA are given respectively in Tables 14 and 15.

[ax,bx]	N_Q	n_{eff}	Res	f_{cost}
[0.98,0.99]	10	0.9807444005 + 7.8735 10^{-5} j	3.06 10^{-2}	1.72 10^{-4}
		0.9820495403 + 1.7631 10^{-7} j	1.85 10^{-1}	7.75 10^{-4}
		0.9845159786 + 3.3767 10^{-8} j	9.19 10^{-1}	1.07 10^{-1}
		0.9845160397+ 4.2854 10^{-8} j	7.31 10^{-1}	9.72 10^{-2}
		0.9887130625 + 6.3461 10^{-6} j	9.83 10^{-2}	9.92 10^{-4}
		0.9887139817 + 6.3485 10^{-6} j	6.43 10^{-2}	6.29 10^{-4}

Table 14: Solutions obtained by CIA with a circle

Note that we do not run several simulations in this case but we will run a second simulation that can be viewed as the second step of an iterative process. The result of the second step will be taken as the reference solution. It results from this that the accuracy of the RIA is strongly efficient in comparison with the CIA (that is nevertheless a good approach).

Knowledge of this first step enables us to focus on a smaller search interval in a second solving step as in Tables 16 and 17. Note that we also take a smaller number of quadrature/interpolation points: 10 for the first step and 5 for the second.

As a result, the iterative application of the process (but with only 2 iterations) achieves a very accurate solution with both methods, but with

[ax,bx]	N_Q	n_{eff}	Res	f_{cost}
[0.98,0.99]	10	0.9807444006 + 7.8735 10^{-5} j	1.98 10^{-4}	3.68 10^{-6}
		0.9820495399 + 1.7616 10^{-7} j	4.57 10^{-4}	1.18 10^{-5}
		0.9845159835 + 3.4127 10^{-8} j	8.66 10^{-4}	3.00 10^{-5}
		0.9845160412+ 3.4109 10^{-8} j	2.70 10^{-3}	6.99 10^{-4}
		0.9887130627 + 6.3466 10^{-6} j	4.14 10^{-5}	6.24 10^{-7}
		0.9887139820 + 6.3483 10^{-6} j	1.29 10^{-4}	8.87 10^{-7}

Table 15: Solutions obtained by RIA

[ax,bx]	N_Q	n_{eff}	Res	f_{cost}
[0.984515,0.984517]	5	0.984515983503 + 3.4129 10^{-8} j	1.02 10^{-7}	9.28 10^{-9}
		0.984516041295 + 3.4101 10^{-8} j	1.63 10^{-7}	3.71 10^{-8}

Table 16: Solutions obtained by CIA with a circle

[ax,bx]	N_Q	n_{eff}	Res	f_{cost}
[0.984515,0.984517]	5	0.984515983503 + 3.4129 10^{-8} j	6.47 10^{-8}	1.39 10^{-8}
		0.984516041295 + 3.4101 10^{-8} j	6.68 10^{-8}	8.60 10^{-9}

Table 17: Solutions obtained by RIA

greater efficiency for RIA (see Table 15) compared to CIA on a circle (see Table 14). This second step may be repeated for each root if needed.

In terms of computation times, this example is more expensive. It takes around 1200 sec for the first step and the small number of quadrature/interpolation points used in the second iteration reduce this to 500 sec. The additionnal cost is essentially due to the increasing time for larger linear systems that have to be solved.

Some remarks can be done

- As the problem size increases, so does the relative contribution of the linear system solver (assembly and solving) to the total computation time.
- For larger systems, it may be worth investigating other integral formulations, particularly to avoid increasing assembly times (which entail additional calculations for the residual).
- Specific acceleration methods such as FMM or H-matrix can be implemented to improve the solving of the linear system.

Note that in this case there is no spurious eigenvalue to eliminate with the computation of the residual (see Step 9 of Algorithm 1).

5. Conclusion

In this work we have compared several approaches for solving the nonlinear eigenvalue problem from the BEM discretization of a PCF problem. Our numerical experiments have shown that methods based on a contour integral are very efficient. Moreover, the comparisons have also allowed us to highlight a greater efficiency of the RIA which, from an algorithmic point of view, is a variant of the contour integral approach. Considered for the first time in this type of application, these methods offer several advantages that allows BEM methods to become efficient. The most important of these are that a starting point close to the solution is not required, and that all solutions for a given interval can be calculated simultaneously. Numerical experiments have also shown that using an iterative process gives better control over the accuracy of the method. Furthermore, as many of the calculations are completely independent, these tools can be easily parallelized.

References

[1] P. Russel, "Photonic crystal fibers," *Science* , Vol. 299, No. 5605, 358–362, January 2003.

[2] Maier, Stefan A and others, "Plasmonics: fundamentals and applications", Vol. 1. New York: springer, 2007.

[3] Y. P. Chiou, Y. C. Chiang, C. H. Lai, C. H. Du, and H. C. Chang, "Finite difference modeling of dielectric waveguides with corners and slanted facets," *Journal of Lightwave Technology*, Vol. 27, No. 12, 2077–2086, December 2009.

[4] F. Brechet, J. Marcou, D. Pagnoux, and P. Roy, " Complete analysis of the characteristics of propagation into photonic crystal fibers by the finite element method," *Optical Fiber Technology*, Vol. 6, 181–191, April 2000.

[5] S. Selleri, L. Vincetti, L. A. Cucinotta, and M. Zoboli, "Complex FEM modal solver of optical waveguides with PML boundary conditions," *Optical Quantum Electronics* , Vol. 33, 359–371, 2001.

[6] H. Cheng, W. Y. Crutchfield, M. Doery, and L. Greengard, "Fast, accurate integral equation methods for the analysis of photonic crystal fibers I: Theory," *Opt. Express*, Vol. 12, 3791–3805, 2004.

[7] Elio Pone, Alireza Hassani, Suzanne Lacroix, Andrei Kabashin, and Maksim Skorobogatiy, "Boundary integral method for the challenging problems in bandgap guiding, plasmonics and sensing," *Opt. Express*, Vol. 15, 10231–10246, 2007.

[8] W. Lu and Y. Y. Lu,outline "Efficient Boundary Integral Equation Method for Photonic Crystal Fibers," *Journal of Lightwave Technology*, Vol. 30, No. 11, 1610–1616, June 2012.

[9] W. Lu and Y. Y. Lu, "Efficient High Order Waveguide Mode Solvers Based on Boundary Integral Equations," *Journal of Computational Physics*, Vol. 272, 507–525, April 2014.

[10] Jun Lai and Shidong Jiang "Second Kind Integral Equation Formulation for the Mode Calculation of Optical Waveguides", *Applied and Computational Harmonic Analysis*, Volume 44, Issue 3, pp 645-664, May 2018.

[11] Haifeng Gao, Toshiro Matsumoto, Toru Takahashi and Hiroshi Isakari, "Eigenvalue analysis for acoustic problem in 3D by boundary element method with the block Sakurai–Sugiura method". *Engineering Analysis with Boundary Elements*, Vol. 37, no. 6: 914–923, 2013.

[12] Mohamed El-Guide, Agnieszka Miedlar, Yousef Saad, "A rational approximation method for solving acoustic nonlinear eigenvalue problems". *Engineering Analysis with Boundary Elements*, Vol. 111: 44–54, 2020.

[13] Wolf-Jürgen Beyn,"An integral method for solving nonlinear eigenvalue problems". *Linear Algebra and Its Applications*, 436(10):3839–3863, 2012.

[14] J. Xiao, C. Zhang, T. M. Huang, and T. Sakurai, "Solving large-scale nonlinear eigenvalue problems by rational interpolation approach and resolvent sampling based Rayleigh-Ritz method," *Int. J. for Numerical Methods in Engineering*, Vol. 110, No. 8, 776–800, 2017.

[15] J.-R. Poirier, J. Vincent, P. Daquin, R. Perrussel and H.-C. Seat "H -Matrix Solver for the Acceleration of Boundary Integral Equation for Photonic Crystal Fiber" *Progress In Electromagnetics Research M* , Vol. 119, 25-35, 2023

[16] David E. Muller, "A Method for Solving Algebraic Equations Using an Automatic Computer". *Mathematical Tables and Other Aids to Computation*, Vol. 10, no. 56: 208–15, 1956.

[17] Min Hyung Cho, Cai Her, Tsing-Hua Her, "A Boundary Integral Equation Method for Photonic Crystal Fibers" *Journal of Scientific Computing*, Vol. 28,263–278, 2006.

[18] Stefan Güttel, and Françoise Tisseur, "The nonlinear eigenvalue problem," *Acta Numerica*, Vol. 26, 1–94, ,Cambridge University Press 2017.

[19] Ömer Eğecioğlu, and Çetin K. Koç, "A fast algorithm for rational black via orthogonal polynomials." *Mathematics of computation* Vol. 53., No. 187, 249-264, 1989.