

EDISON

Electromagnetic Design of
flexIble SensOrs



Report 94 fast CISS for Dispersive Surface Boundary Conditions

dr eng. Grzegorz Fotyga
October 11, 2021



"EDISON–Electromagnetic Design of flexIble SensOrs" Project, Agreement
POIR.04.04.00-00-1DC3/16-00 date December 6, 2016, within the TEAM-TECH Program of the
Foundation for Polish Science co-financed by the European Union under the European Regional
Development Fund.

a A contour integral-based projection method for nonlinear eigenproblems associated with dispersive boundary conditions

Assuming the dispersive surface boundary conditions of the problem the finite element formulation (FEM) leads to the following **nonlinear** eigenproblem (NEP):

$$(\Gamma + \gamma(s)G + s^2C)e = 0. \quad (1)$$

where s is a frequency parameter¹, $s \in \Omega$, and $\Omega \subset \mathbb{C}$. $\gamma(s) \sim \sqrt{s}$ is a scalar function modeling the dispersion of the surface impedance. FEM system matrices: $\Gamma, G, C \in \mathbb{R}^{n \times n}$, n is the number of unknowns, $e \in \mathbb{C}^n$, $e \neq 0$. Eq. (1) can be represented as follows:

$$T(s)e = 0. \quad (2)$$

Here, we assume that $T(s) : \Omega \rightarrow \mathbb{C}^{n \times n}$ is holomorphic in domain Ω .

The goal is to find all eigenvalues that lie in the closed contour Γ and associated eigenvectors using the method proposed in:

1. *Yokota, Shinnosuke, and Tetsuya Sakurai. "A projection method for nonlinear eigenvalue problems using contour integrals." Jsiam Letters 5 (2013): 41-44.*

To this end one has to generate an eigenspace using contour integrals for curve Γ :

$$S_\kappa = \frac{1}{2\pi i} \int_\Gamma g_\kappa(s)T(s)^{-1}U ds, \quad \kappa \in \mathbb{N} \quad (3)$$

where $U \in \mathbb{C}^{n \times L}$, $L \leq n$ and g_κ are polynomial functions of κ -th degree.

Next, we evaluate the contour integral (3) using a numerical integral, resulting in:

$$\hat{S}_\kappa = \sum_{p=1}^N w_p g_\kappa(s_p)T(s_p)^{-1}U, \quad (4)$$

where s_p are integral points for $p = 1, \dots, N$, w_p are weights corresponding to s_p . It has to be noted that this step is the most time consuming, since it requires to solve N system of equations with large number (n) of unknowns. Once all matrices \hat{S}_κ are computed for $\kappa = 0, \dots, K - 1$ one can construct the orthonormal projection basis:

$$\text{span}(Q) = \text{span}([\hat{S}_0 \dots \hat{S}_{K-1}]) \quad (5)$$

and use it to generate the reduced eigenproblem by means of e.g. Rayleigh-Ritz subspace projection technique (see [1] for details).

b Inhouse CISS results

Here, we use the method described in **Section a** to compute the resonant frequencies of the structure shown in Fig. 1.

- Resonator ($22.86 \times 10.16 \times 30$ mm) with dispersive surface boundary conditions: $\sigma = 5800$
- Problem size = 362678
- integration points: $N = 80$
- size of left/right probing space made of random vectors, denoted by U : 11

¹ $s = j\omega/c = jk_0$

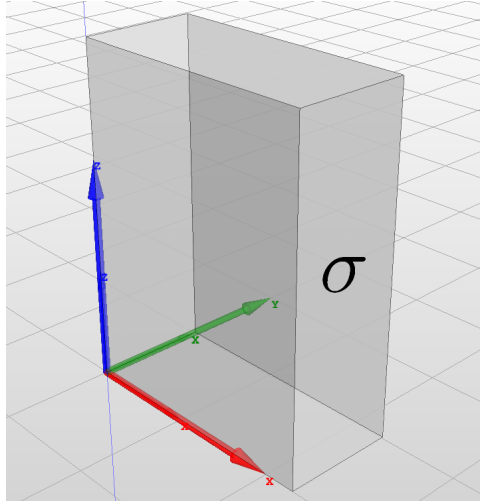


Figure 1: Resonator with dispersive surface boundary conditions.

- number of moments: 2
- computational time: 1677.6s

l	$\ \mathbf{T}(k) \cdot \mathbf{x}\ / k $	f[GHz]
1	1e-13	8.190395959928 + 0.052963326912i
2	1e-13	11.890412063906 + 0.061543106886i
3	1e-12	13.962144599581 + 0.071396538013i
4	1e-13	15.483001263886 + 0.093268182875i
5	1e-12	16.056033990150 + 0.088795266000i
6	1e-13	16.289902974497 + 0.071046891862i
7	1e-12	16.412135641333 + 0.075439113972i
8	1e-13	16.780801725899 + 0.119114475797i
9	1e-13	16.796993129566 + 0.103175257975i

c FAST CISS idea

In this section the modification of the standard CISS method is proposed which leads to the speed up of computations without devoting the computational accuracy. Originally, in order to construct the projection basis Q , one has to solve large system of equations N times (4) using e.g. LU factorization. This is definitely the most time consuming step in the whole process. The solution vectors, which are used to construct Q in (4) and (5), can be gathered together, creating the following matrix:

$$E = [E(s_1), E(s_2) \dots E(s_N)] \quad (6)$$

where

$$T(s_i)E(s_i) = U \quad (7)$$

and $E \in \mathbb{C}^{n \times NL}$. However, usually the rank of E is much lower than the number of columns (NL). In other words, there is a redundancy in the basis, since the electromagnetic field does not vary significantly inside a given frequency band of interest. Our goal is to create the reduced basis \hat{V} , which approximates E with sufficient accuracy:

$$\text{span}(\hat{V}) \approx \text{span}(E) \quad (8)$$

Here, \hat{V} is composed of solution vectors $[E(\hat{s}_1), E(\hat{s}_2) \dots E(\hat{s}_{\hat{N}})]$, computed in properly specified set of \hat{N} frequency points $\hat{s}_1, \hat{s}_2, \dots, \hat{s}_{\hat{N}}$, where usually $\hat{N} \ll N$. Thus, the whole process is performed much faster. The orthogonality of the projection basis \hat{V} is ensured by the Gram–Schmidt process. Applying the Galerkin method with \hat{V} as basis and testing vectors, a reduced order model (ROM) system is obtained:

$$\hat{T}(s)\hat{E}(s) = \hat{U} \quad (9)$$

where

$$\hat{T}(s) = \hat{\Gamma} + \gamma(s)\hat{G} + s^2\hat{C} \quad (10)$$

and the reduced matrices are given by: $\hat{\Gamma} = \hat{V}^H \Gamma \hat{V} \in \mathbb{C}^{R \times R}$, $\hat{G} = \hat{V}^H G \hat{V} \in \mathbb{C}^{R \times R}$, $\hat{C} = \hat{V}^H C \hat{V} \in \mathbb{C}^{R \times R}$, $\hat{U}_r = \hat{V}^H U \in \mathbb{C}^{R \times L}$. Note that $R \ll NL \ll n$

Each time a new solution matrix $E(\hat{s}_i)$ is appended to the projection basis \hat{V} , the error introduced by the ROM is assessed by means of a correctly defined error estimator. The role of the estimator is twofold: Firstly, it is used as a stopping criterion—once the estimated error drops below a specified tolerance over the entire frequency band, the algorithm stops. Secondly, it guides the process of selecting expansion frequency points within the band of interest. To this end, a greedy scheme is applied. That is, the next point is located at the frequency at which the estimated error is largest. It is worth mentioning that this kind of reduction scheme based on the error-estimator enables full automation of the reduction process, since at every stage of the reduction, we gain insight into the error introduced by the ROM.

In order to define the error estimator formula, we first have to formulate the residual error matrix:

$$R(s) = U - (\Gamma + \gamma(s)G + s^2C)\hat{V}\hat{E}(s) \quad (11)$$

where the reduced solution matrix $\hat{E}(s)$ is the solution of (9).

Based on (9) and (11) we can efficiently assess the error introduced by the reduced order model using the following goal-oriented error estimator:

$$err_{est}(s) = \|R^T R\| / \|U^T U\| \quad (12)$$

Note that the blocks of matrices in expanded version of (12) are frequency-independent and can be computed only once, in the preprocessing stage, so-called *offline*. More precisely, the offline stage comprises all the computations with complexity depending on n , which are performed only once. On the other hand, the *online* phase concerns all the computations with the complexity depending on R , including error estimation process (12) and fast frequency sweep (9).

The above definition of the error estimator is used to fully control the reduction algorithm in terms of the stopping criteria, as well as the expansion point selection. Although the estimator operates on low-order matrices, it must be computed at each frequency within the band of interest, every time a projection basis is expanded.

Finally, the blocks of matrices used in (6) are approximated using:

$$E(s) \approx \hat{V}\hat{E}(s). \quad (13)$$

To sum up, \hat{V} is usually computed much faster as compared to E and it allows one to obtain all eigenpairs with sufficient accuracy.

d Fast CISS tests

- Experiment 1

– $tol = 1e-3$

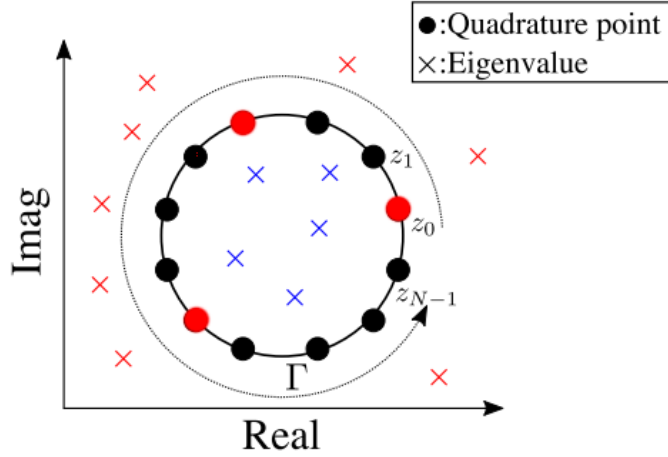


Figure 2: Quadrature points selected using **greedy** strategy.

- integration points: $nc = 80$ (7 selected)
- size of left/right probing space (random vectors): 11
- number of moments: 2
- computational time: 199.8s

1	$\ T(k) \cdot x\ / k $	f[GHz]
1	1e-6	8.190396680354 + 0.052963179817i
2	1e-4	11.890411888083 + 0.061535579489i
3	1e-5	13.962147346702 + 0.071396493527i
4	1e-5	15.483002065627 + 0.093269193525i
5	1e-6	16.056034069277 + 0.088795245148i
6	1e-6	16.289902939957 + 0.071046852259i
7	1e-6	16.412135432651 + 0.075439258222i
8	1e-6	16.780801643546 + 0.119114440769i
9	1e-5	16.796993180353 + 0.103175183494i

- Experiment 2

- $tol = 1e-5$
- integration points: $nc = 80$ (9 selected)
- size of left/right probing space (random vectors): 11
- number of moments: 2
- computational time: 265.2s

1	$\ T(k) \cdot x\ / k $	f[GHz]
1	1e-7	8.190396024796 + 0.052963278720i
2	1e-6	11.890411990728 + 0.061543174083i
3	1e-6	13.962144506062 + 0.071396503104i
4	1e-6	15.483001106192 + 0.093268145372i
5	1e-6	16.056033971201 + 0.088795389806i
6	1e-6	16.289902976252 + 0.071046897588i
7	1e-7	16.412135664158 + 0.075439091585i
8	1e-7	16.780801710781 + 0.119114481450i
9	1e-7	16.796993067579 + 0.103175293053i

Experiment 3

- $tol = 1e-6$
- integration points: $nc = 80$ (9 selected)
- size of left/right probing space (random vectors): 21
- number of moments: 2
- computational time: 350.4s

l	$ T(k) \cdot x / k $	f[GHz]
1	1e-9	8.190395958774 + 0.052963327342i
2	1e-8	11.890412062933 + 0.061543107206i
3	1e-8	13.962144599058 + 0.071396533864i
4	1e-8	15.483001264349 + 0.093268179928i
5	1e-9	16.056033990421 + 0.088795266265i
6	1e-9	16.289902974774 + 0.071046891999i
7	1e-9	16.412135641443 + 0.075439115063i
8	1e-9	16.780801725462 + 0.119114476070i
9	1e-8	16.796993129555 + 0.103175258467i

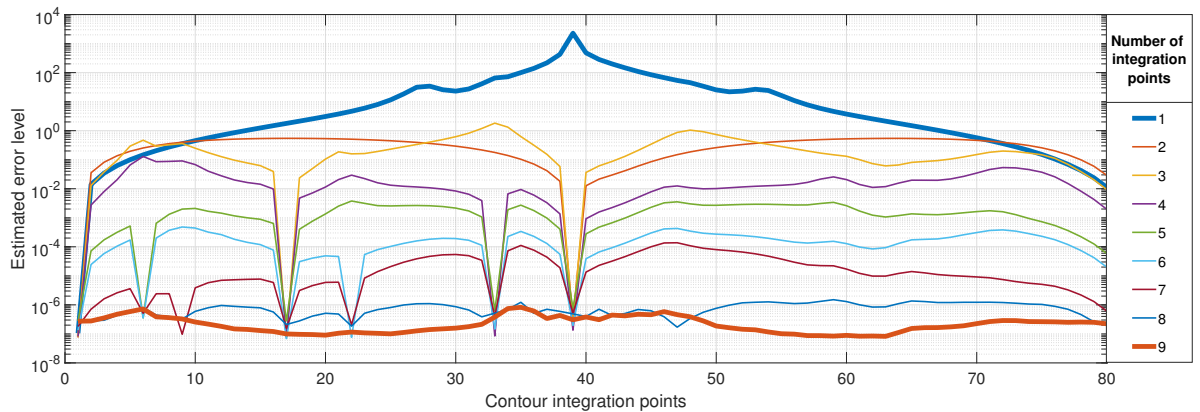


Figure 3: Estimated error for 1-9 integration points.