Calculation of Eigenmodes for Accelerator Cavities with Losses using CIM

V. Pham-Xuan, W. Ackermann and H. De Gersem

Institut für Theorie Elektromagnetischer Felder
Outline of the Talk

Motivation
  Iterative methods
  Contour integral methods

Formulation
  Mathematical model
  Contour integral methods
  Multigrid method as a preconditioner
  Block Krylov methods and recycling Krylov methods

Numerical Results

Possible Improvements
Motivation
  Iterative methods
  Contour integral methods

Formulation
  Mathematical model
  Contour integral methods
  Multigrid method as a preconditioner
  Block Krylov methods and recycling Krylov methods

Numerical Results

Possible Improvements
Motivation

Problem statement

Problem statement: we have to solve a **nonlinear eigenvalue problem (NEP)** where

- the problem is **large and sparse**;
- the number of eigenvalues is large;
- **prior information** about eigenvalues is available;
- in several applications, one is only interested in a **few eigenvalues within a certain range**.

**Figure:** Chain of cavities (from [1])

Available methods:

- Iterative methods: Jacobi-Davidson [2], Arnoldi, Lanczos, etc.
- Contour integral methods: Beyn methods [3], resolvent sampling based Rayleigh-Ritz method (RSRR) [4], etc.
Motivation
  Iterative methods
  Contour integral methods

Formulation
  Mathematical model
  Contour integral methods
  Multigrid method as a preconditioner
  Block Krylov methods and recycling Krylov methods

Numerical Results

Possible Improvements
Motivation
Iterative methods (Jacobi-Davidson)

Lossy accelerator cavity: eigenvalues are in the complex plane

- choose an initial guess
- expand the search space ...
- until an approximate solution is found
Lossy accelerator cavity: eigenvalues are in the complex plane

- choose an initial guess
- expand the search space ...
- until an approximate solution is found
- choose another initial guess
- continue expanding the search space ...
- find another approximate solution
Motivation
Iterative methods (Jacobi-Davidson)

Lossy accelerator cavity: eigenvalues are in the complex plane

• choose an initial guess
• expand the search space ...
• until an approximate solution is found

• if we choose unsuitable initial guess
• the algorithm will converge to ...
• a previously determined eigenvalue!!!!!
Presentation Outline

Motivation
  Iterative methods
  Contour integral methods

Formulation
  Mathematical model
  Contour integral methods
  Multigrid method as a preconditioner
  Block Krylov methods and recycling Krylov methods

Numerical Results

Possible Improvements
Motivation
Contour integral methods

An accurate computation of eigenpairs inside a region enclosed by a non-self-intersecting curve.

- choose a region to look for eigenvalues
- the region can be of any shape, e.g rectangle, circle/ellipse.
An accurate computation of eigenpairs inside a region enclosed by a non-self-intersecting curve.

- choose a region to look for eigenvalues
- the region can be of any shape, e.g. rectangle, circle/ellipse.
- most computation is spent to solve linear equation systems at different interpolation points, which can be parallelized.
Motivation
- Iterative methods
- Contour integral methods

Formulation
- Mathematical model
- Contour integral methods
- Multigrid method as a preconditioner
- Block Krylov methods and recycling Krylov methods

Numerical Results

Possible Improvements
Presentation Outline

Motivation
- Iterative methods
- Contour integral methods

Formulation
- Mathematical model
- Contour integral methods
- Multigrid method as a preconditioner
- Block Krylov methods and recycling Krylov methods

Numerical Results

Possible Improvements
The combination of Maxwell-Ampère equation and the Maxwell-Faraday equation results in the double-curl equation

\[ \nabla \times \frac{1}{\mu} \nabla \times \vec{E} + j\omega \sigma \vec{E} = \varepsilon \omega^2 \vec{E} \]  

(1)

Applying the Galerkin's approach to discretize (1) results in an eigenvalue problem

\[ A^{3D}x + j\omega \mu_0 C^{3D}x - \omega^2 \mu_0 \varepsilon_0 B^{3D}x = 0 \]  

(2)

which includes only losses from volumetric lossy material. Special treatment is carried out to incorporate 2D losses at port interfaces into (2), resulting in a nonlinear eigenvalue problem (NEP)

\[ P(z)x = 0 \]  

(3)

where \( z = \left( \frac{\omega}{2\pi f_\tau} \right)^2 \)
Presentation Outline

Motivation
- Iterative methods
- Contour integral methods

Formulation
- Mathematical model
- Contour integral methods
- Multigrid method as a preconditioner
- Block Krylov methods and recycling Krylov methods

Numerical Results

Possible Improvements
To solve $P(z)x = 0$ most of the standard eigenvalue algorithms exploit a projection procedure in order to extract approximate eigenvectors from a given subspace.

1. Construct an approximate eigenspace $Q$
2. Obtain matrix $Q$
3. Project the original NEP to a reduced NEP using Rayleigh-Ritz procedure:
   
   $P_Q(z) = Q^H P(z) Q$
4. Solve the reduced NEP
   
   $P_Q(z)g = 0$
5. Compute eigenpairs of the original NEP
   
   same eigenvalues as for the reduced problem; eigenvectors $x = Qg$
The evaluation of $Q$ requires the computation of

$$\frac{1}{2\pi i} \oint_{\Gamma} P(z)^{-1} V dz$$  \hspace{1cm} (4)$$

$P(z_n)$ is the matrix system at an integration point $z_n$. $V$ is a random matrix.

The most expensive operation is to compute

$$X = P^{-1}(z_i) V$$  \hspace{1cm} (5)$$
Presentation Outline

Motivation
- Iterative methods
- Contour integral methods

Formulation
- Mathematical model
- Contour integral methods
- Multigrid method as a preconditioner
- Block Krylov methods and recycling Krylov methods

Numerical Results

Possible Improvements
The most expensive operation is to compute

\[ X = P^{-1}(z_i)V \]  

\[ (6) \]

equivalent to solving the linear system

\[ P(z_i)X = V \]  

\[ (7) \]

- Direct inverse becomes prohibitively expensive for large problems.
- For large-scale problems, iterative methods are preferable.
- Linear systems generated by Maxwell’s equations are extremely ill-conditioned.
- Krylov iterative solvers with simple preconditioners often stagnate or diverge when applied to such linear systems.
- **Suitable preconditioners/iterative solvers** should be applied to improve the convergence of the iterative solvers.
Formulation
Multigrid method as a preconditioner

Formulation
Multigrid method as a preconditioner

\[ M^{-1} b = e \]  
\( (8) \)

This equation is repeatedly computed at each iteration where \( M \) is the preconditioner, \( b \) is the input and \( e \) is the output. The output is computed by solving systems of the type

\[ P_{ii} e_i = b_i - \sum_{i \neq j} P_{ij} e_j \]  
\( (9) \)

where \( i \) and \( j \) refer to the order of the trial and test functions.

Presentation Outline

Motivation
- Iterative methods
- Contour integral methods

Formulation
- Mathematical model
- Contour integral methods
- Multigrid method as a preconditioner
- Block Krylov methods and recycling Krylov methods

Numerical Results

Possible Improvements
At each integration point, we need to solve a system with multiple right-hand-sides (RHSs)
At each integration point, we need to solve a system with multiple right-hand-sides (RHSs).

We sequentially solve for each RHS. Each solution is approximated from a Krylov space $x_i \in \mathcal{K}(P, v_i)$.
At each integration point, we need to solve a system with multiple right-hand-sides (RHSs)

We sequentially solve for each RHS. Each solution is approximated from a Krylov space $x_i \in \mathcal{K}(P, v_i)$
### Formulation

**Block Krylov methods**

At each integration point, we need to solve a system with multiple right-hand-sides (RHSs).

We sequentially solve for each RHS. Each solution is approximated from a Krylov space
\[ x_i \in \mathcal{K}(P, v_i) \]

The block Krylov methods solve for all RHSs at once. Each solution is approximated from a larger subspace
\[ \mathcal{B} = \mathcal{K}(P, v_1) + \ldots + \mathcal{K}(P, v_n) \]
Formulation
Recycling Krylov methods

\[ \begin{array}{ccc}
P(z) & X & V \\
\cdot & = & \\
\end{array} \]

Approximate the first solution from \( \mathcal{K}(P, v_1) \)
Formulation
Recycling Krylov methods

Approximate the first solution from $\mathcal{K}(P, v_1)$

Approximate the second solution from $\mathcal{R}^{(1)} + \mathcal{K}(P, v_2)$
Formulation
Recycling Krylov methods

Approximate the first solution from $\mathcal{K}(P, v_1)$

Approximate the second solution from $\mathcal{R}^{(1)} + \mathcal{K}(P, v_2)$

Approximate the third solution from $\mathcal{R}^{(2)} + \mathcal{K}(P, v_3)$
Presentation Outline

Motivation
  - Iterative methods
  - Contour integral methods

Formulation
  - Mathematical model
  - Contour integral methods
  - Multigrid method as a preconditioner
  - Block Krylov methods and recycling Krylov methods

Numerical Results

Possible Improvements
Numerical Results
Accuracy Validation

- Name: spherical cavity
- Electrical conductivity: $62 \times 10^6$ S/m
- Radius: 1m
- Degrees of freedom: 7914/18728/36730/98244/205056
- Target frequency: 125MHz
- Number of eigenvalues: 3
- Algorithm parameters:
  - Region: rectangle$(1.0, 1.5, 1.0 \times 10^{-15}, 0.05)$
  - $N = 20$ (number of integration points)
  - $L = 20:10:100$ (number of columns of the random matrix)
  - $K = 2$ (for BEYN2)
  - Rank tolerance = $1.0 \times 10^{-10}$
Numerical Results

Accuracy Validation

\[ \epsilon_f = \frac{\| f - f_{\text{analytical}} \|}{\| f_{\text{analytical}} \|} \]

\[ \epsilon_Q = \frac{\| Q - Q_{\text{analytical}} \|}{\| Q_{\text{analytical}} \|} \]
Numerical Results
Incorporating losses from ports

- Name: spherical cavity with a coupler
- Radius: 1m
- Degrees of freedom: 3294/6158/10854/17880
- Target frequency: 125MHz
- Number of eigenvalues: 3
- Algorithm parameters:
  - Region: rectangle(1.0, 1.5, 1.0 × 10^{-15}, 0.05)
  - N = 20 (number of integration points)
  - L = 20:10:100 (number of columns of the random matrix)
  - K = 2 (for BEYN2)
  - Rank tolerance = 1.0 × 10^{-10}
Numerical Results
Incorporating losses from ports

<table>
<thead>
<tr>
<th>DOF</th>
<th>3294</th>
<th>6158</th>
<th>10854</th>
<th>17880</th>
</tr>
</thead>
<tbody>
<tr>
<td>Freq. 1 (MHz)</td>
<td>130.641496</td>
<td>130.579307</td>
<td>130.543168</td>
<td>130.561674</td>
</tr>
<tr>
<td>Freq. 2 (MHz)</td>
<td>130.810936</td>
<td>130.756186</td>
<td>130.706348</td>
<td>130.698581</td>
</tr>
<tr>
<td>Freq. 3 (MHz)</td>
<td>130.821949</td>
<td>130.760146</td>
<td>130.724020</td>
<td>130.699990</td>
</tr>
<tr>
<td>Qual. 1</td>
<td>2.771 842 × 10^2</td>
<td>2.469 505 × 10^2</td>
<td>2.787 006 × 10^2</td>
<td>2.871 997 × 10^2</td>
</tr>
<tr>
<td>Qual. 2</td>
<td>2.716 048 × 10^5</td>
<td>2.875 342 × 10^6</td>
<td>2.192 351 × 10^6</td>
<td>1.114 248 × 10^8</td>
</tr>
<tr>
<td>Qual. 3</td>
<td>2.109 400 × 10^6</td>
<td>2.836 039 × 10^7</td>
<td>7.579 235 × 10^6</td>
<td>1.663 706 × 10^7</td>
</tr>
</tbody>
</table>

Table: Resonant frequencies and quality factors of spherical cavity with a coupler for different mesh sizes
Numerical Results
Recycling Subspace Methods

Using recycling subspace methods provided by a framework for high-performance domain decomposition methods (HPDDM) [5]

- Name: spherical cavity with a coupler
- Radius: 1m
- Degrees of freedom: 17880
- Number of RHSs: 20
- Tolerance: $10 \times 10^{-9}$

<table>
<thead>
<tr>
<th>Krylov Methods</th>
<th>GMRES</th>
<th>GCRODR-5</th>
<th>GCRODR-10</th>
<th>GCRODR-20</th>
<th>BGMRES</th>
<th>BGCRODR</th>
</tr>
</thead>
<tbody>
<tr>
<td>average # of iterations</td>
<td>29</td>
<td>25.2</td>
<td>25.1</td>
<td>24.3</td>
<td>24</td>
<td>24</td>
</tr>
<tr>
<td>runtime (s)</td>
<td>18.4</td>
<td>22.5</td>
<td>20.5</td>
<td>20.5</td>
<td>19</td>
<td>23.8</td>
</tr>
</tbody>
</table>

**Table:** Number of iterations and runtime for different Krylov methods
Numerical Results

Recycling Subspace Methods

Using recycling subspace methods provided by a framework for high-performance domain decomposition methods (HPDDM) [5]

- Name: Tesla cavity with a main coupler
- Degrees of freedom: 60970
- Number of RHSs: 20
- Tolerance: $10 \times 10^{-9}$

<table>
<thead>
<tr>
<th>Krylov Methods</th>
<th>GMRES</th>
<th>GCRODR-5</th>
<th>GCRODR-10</th>
<th>GCRODR-20</th>
<th>BGMRES</th>
<th>BGCRODR</th>
</tr>
</thead>
<tbody>
<tr>
<td>average # of iterations</td>
<td>31.7</td>
<td>26.4</td>
<td>26.3</td>
<td>25.9</td>
<td>25</td>
<td>25</td>
</tr>
<tr>
<td>runtime (s)</td>
<td>69</td>
<td>63.3</td>
<td>76.8</td>
<td>61.2</td>
<td>89.2</td>
<td>78.9</td>
</tr>
</tbody>
</table>

**Table:** Number of iterations and runtime for different Krylov methods
Presentation Outline

Motivation
  Iterative methods
  Contour integral methods

Formulation
  Mathematical model
  Contour integral methods
  Multigrid method as a preconditioner
  Block Krylov methods and recycling Krylov methods

Numerical Results

Possible Improvements
Possible Improvements

- Improve the speed of exporting system matrices.
Possible Improvements

- Improve the speed of exporting system matrices.
- Improve the efficiency in using block iterative methods.
Possible Improvements

- Improve the speed of exporting system matrices.
- Improve the efficiency in using block iterative methods.
- Apply the Auxiliary Space Preconditioning [6, 7, 8] to reduce the cost in solving iterative problems.
Thank you for your attention


Appendix
Iterative methods (Jacobi-Davidson)

Lossless accelerator cavity: eigenvalues are on real axis

- choose an initial guess
- expand the search space ...
- until an approximate solution is found
Appendix
Iterative methods (Jacobi-Davidson)

Lossless accelerator cavity: eigenvalues are on real axis

- choose an initial guess
- expand the search space ...
- until an approximate solution is found
- the solution becomes the new initial guess
- continue expanding the search space ...
The resolvent $P(z)^{-1}$ reveals the existence of eigenvalues, indicates where eigenvalues are located, and show how sensitive these eigenvalues are to perturbation.

As explained in [3], from Keldysh’s theorem, we know that the resolvent function $P(z)^{-1}$ can be written (for simple eigenvalues $\lambda_i$) as

$$P(z)^{-1} = \sum_i v_i w_i^H \frac{1}{z - \lambda_i} + R(z) \quad (10)$$

where

- $v_i$ and $w_i$ are suitably scaled right and left eigenvectors, respectively, corresponding to the (simple) eigenvalue $\lambda_i$
- $R(z)$ and $P(z)$ are analytic functions
Appendix

Contour integral methods

Some basic spectral theory

$$Q = (q_1, q_2, \ldots, q_k)$$

$$\text{span}\{q_1, q_2, \ldots, q_k\} \supseteq \text{span}\{v_1, v_2, \ldots, v_{n(\Gamma)}\}$$

$$P(z)^{-1} = \sum_i v_i w_i^H \frac{1}{z - \lambda_i} + R(z)$$

Applying Cauchy’s integral formula

$$\frac{1}{2\pi i} \oint_{\Gamma} f(z) P(z)^{-1} dz = \sum_{i=1}^{n(\Gamma)} f(\lambda_i) v_i w_i^H$$

In practice, we evaluate the integral

$$\frac{1}{2\pi i} \oint_{\Gamma} f(z) P(z)^{-1} \hat{V} dz$$
Appendix

Contour integral methods

Some basic spectral theory

\[ P(z)^{-1} = \sum_{i} v_i w_i^H \frac{1}{z - \lambda_i} + R(z) \]

Applying Cauchy’s integral formula

\[ \frac{1}{2\pi i} \oint f(z) P(z)^{-1} dz = \sum_{i=1}^{n(\Gamma)} f(\lambda_i) v_i w_i^H \]

In practice, we evaluate the integral

\[ \frac{1}{2\pi i} \oint f(z) P(z)^{-1} \hat{V} dz \]

Using interpolation, we obtain

\[ \frac{1}{2\pi i} \oint f(z) P(z)^{-1} \hat{V} dz = \sum_{i=1}^{n_{int}} \xi_i f(z_i) P(z_i)^{-1} \hat{V} \]

\[ Q = (q_1, q_2, ..., q_k) \]

\[ \text{span}\{q_1, q_2, ..., q_k\} \supseteq \text{span}\{v_1, v_2, ..., v_{n(\Gamma)}\} \]
Appendix

Contour integral methods

Beyn1 (for a few eigenvalues)

Define the matrices $A_0$ and $A_1 \in \mathbb{C}^{n \times k}$

\[ A_0 = \frac{1}{2\pi i} \oint_{\Gamma} P(z)^{-1} \hat{V} dz \] (11)

\[ A_1 = \frac{1}{2\pi i} \oint_{\Gamma} zP(z)^{-1} \hat{V} dz \] (12)

Then $A_0 = VW^H \hat{V}$ and $A_1 = V \Lambda W^H \hat{V}$ where

- $\Lambda = \text{diag}(\lambda_1, ..., \lambda_{n(\Gamma)})$
- $V = \begin{bmatrix} v_1 & \cdots & v_{n(\Gamma)} \end{bmatrix}$
- $W = \begin{bmatrix} w_1 & \cdots & w_{n(\Gamma)} \end{bmatrix}$

$\hat{V}$ is a random matrix $\hat{V} \in \mathbb{C}^{n \times L}$. $L$ is smaller than $n$ and equal or greater and $k$
Appendix

Contour integral methods

Beyn1 (for a few eigenvalues)

Beyn’s method is based on the singular value decomposition of $A_0$

$$A_0 = V_0 \Sigma_0 W_0^H$$  \hspace{1cm} (13)

Beyn has shown that the matrix

$$B = V_0^H A_1 W_0^H \Sigma_0^{-1}$$  \hspace{1cm} (14)

is diagonalizable. Its eigenvalues are the eigenvalues of $P$ inside the contour and its eigenvectors lead to the corresponding eigenvectors of $P$. 
Appendix
Contour integral methods
Beyn2 (for many eigenvalues)

Define the matrices $A_p \in \mathbb{C}^{n \times k}$

$$A_p = \frac{1}{2\pi i} \oint_{\Gamma} z^p P(z)^{-1} \hat{V} dz$$ \hspace{1cm} (15)

Then $A_p = V\Lambda^p W^H \hat{V}$. The matrices $B_0$ and $B_1$ are defined as follows

$$B_0 = \begin{pmatrix} A_0 & \cdots & A_{K-1} \\ \vdots & \ddots & \vdots \\ A_{K-1} & \cdots & A_{2K-2} \end{pmatrix} \hspace{1cm}; \hspace{1cm} B_1 = \begin{pmatrix} A_1 & \cdots & A_K \\ \vdots & \ddots & \vdots \\ A_K & \cdots & A_{2K-1} \end{pmatrix}$$ \hspace{1cm} (16)
Appendix
Contour integral methods
Beyn2 (for many eigenvalues)

Performing the singular value decomposition of $B_0$

$$B_0 = V_0 \Sigma_0 W_0^H$$  \hspace{1cm} (17)

Beyn has shown that the matrix

$$D = V_0^H B_1 W_0^H \Sigma_0^{-1}$$  \hspace{1cm} (18)

is diagonalizable. Its eigenvalues are the eigenvalues of $P$ inside the contour and its eigenvectors lead to the corresponding eigenvectors of $P$. 
Appendix

Contour integral methods

Resolvent Sampling based Rayleigh-Ritz method

The Beyn2 algorithm is robust and accurate if a large $L$ but a small $K$ are used.

However, for large-scale problems, a small $L$ is essential to reduce the computational burden.

Decrease $L$ and increase $K$ make the algorithm unstable and inaccurate.
Appendix

Contour integral methods

Resolvent Sampling based Rayleigh-Ritz method

The Beyn2 algorithm is robust and accurate if a large $L$ but a small $K$ are used.

However, for large-scale problems, a small $L$ is essential to reduce the computational burden.

Decrease $L$ and increase $K$ make the algorithm unstable and inaccurate.

RSRR reduce the number of columns of $V$.

Let $Q \in \mathbb{C}^{n \times k}$ be an orthogonal basis of search space, then the original NEP can be converted to the following reduced NEP

$$P_Q(z)g = 0$$
Appendix

Contour integral methods

Resolvent Sampling based Rayleigh-Ritz method

(1) Initialization: Fix the contour $\Gamma$, the number $N$ and the sampling points $z_i$. Fix the number $L$ and generate a $n \times L$ random matrix $U$

(2) Compute $P(z_i)^{-1}U$ for $i = 0, 1, ..., N - 1$

(3) Form $S$ as follows

$$S = \begin{bmatrix} P(z_0)^{-1}U, & P(z_1)^{-1}U, & \cdots, & P(z_{N-1})^{-1}U \end{bmatrix} \in \mathbb{C}^{n \times N \cdot L} \tag{19}$$

(4) Generate the matrix $Q$ via the truncated singular value decomposition $S \approx Q \Sigma V^H$.

(5) Compute $P_Q(z) = Q^H P(z) Q$, and solve the projected NEP $P_Q(\lambda)g = 0$ using the SS-FULL algorithm to obtain $n(\Gamma)$ eigenpairs $(g_j, \lambda_j)$.

(6) Compute the eigenpairs of the original NEP via the eigenpairs of the reduced NEP.
Appendix
Nonlinear Eigenvalue Solver for Accelerator Cavities (NES4AC)

CST

CEM3D [9]

cavity design, FEM discretization

Solve nonlinear eigenvalue problem

generate matrices $P(z_i)$
Appendix

Nonlinear Eigenvalue Solver for Accelerator Cavities (NES4AC)

Parallell level 1:
- multiple contours

Parallel level 2:
- compute nodes in parallel (implemented in codes)

Parallel level 3:
- parallelize computation at each node (implemented in the codes)

\[ P(\lambda)x = v \]

\[ x = P(z)^{-1}v \]
Appendix

Nonlinear Eigenvalue Solver for Accelerator Cavities (NES4AC)

Parallel level 1:
multiple contours

Parallel level 2:
compute nodes in parallel (implemented)

Parallel level 3:
parallelize computation at each node (implemented in the codes)

\[ P(\lambda)x = v \]

or \[ x = P(z)^{-1}v \]
Appendix

Nonlinear Eigenvalue Solver for Accelerator Cavities (NES4AC)

Parallel level 1: multiple contours

Parallel level 2: compute nodes in parallel (implemented)

Parallel level 3: parallelize computation at each node (implemented in the codes)

\[ P(\lambda)x = v \text{ or } x = P(z)^{-1}v \]

\[ V \oint_{\Gamma} P^{-1}(z)V dz \]
Appendix
Nonlinear Eigenvalue Solver for Accelerator Cavities (NES4AC)

Parallel level 1: multiple contours

Parallel level 2: compute nodes in parallel (implemented)

Parallel level 3: parallelize computation at each node (implemented in the codes)

\[ P(\lambda)x = v \quad \text{or} \quad x = P(z)^{-1}v \]

\[ \oint_{\Gamma} P^{-1}(z)V dz \]
Appendix

Nonlinear Eigenvalue Solver for Accelerator Cavities (NES4AC)

Parallel level 1: multiple contours
Parallel level 2: compute nodes in parallel (implemented)
Parallel level 3: parallelize computation at each node (implemented in the codes)

\[ P^{-1}(\lambda)x = v \]

\[ P^{-1}(z)V \]

\[ \oint_{\Gamma} P^{-1}(z)V \, dz \]
Appendix

Nonlinear Eigenvalue Solver for Accelerator Cavities (NES4AC)

Parallel level 1: multiple contours

Parallel level 2: compute nodes in parallel (implemented)

Parallel level 3: parallelize computation at each node (implemented)

\[ P(\lambda)x = v \]

or

\[ x = P(z)^{-1}v \]
Appendix

Nonlinear Eigenvalue Solver for Accelerator Cavities (NES4AC)

Parallel level 1: multiple contours

Parallel level 2: compute nodes in parallel (implemented)

Parallel level 3: parallelize computation at each node (implemented)

\[ P(\lambda) x = v \]

or \[ x = P(z)^{-1} v \]
Appendix

Nonlinear Eigenvalue Solver for Accelerator Cavities (NES4AC)

NES4AC highlights:

- extends the functionality of CEM3D [9].
- parallelized and developed in C++.
- based on PETSc (Portable, Extensible Toolkit for Scientific Computation) v3.3.0 and LAPACK.
- adopts the parallel scheme of the contour integral method from SLEPc (Scalable Library for Eigenvalue Problem Computations).
- uses the superLU_DIST for the computation of LU decompositions.
- including three contour integral algorithms for eigenvalue solution: Beyn1 (for a few eigenvalues), Beyn2 (for many eigenvalues) and RSRR.
- with two types of closed contour: ellipse and rectangle.
The combination of Maxwell-Ampère equation and the Maxwell-Faraday equation results in the double-curl equation

\[ \nabla \times \left( \frac{1}{\mu} \nabla \times \vec{E} \right) + j\omega \sigma \vec{E} = \varepsilon \omega^2 \vec{E} \]  

(20)

Applying the Galerkin’s approach to discretize (1) results in an eigenvalue problem

\[ A^{3D} x + j \omega \mu_0 C^{3D} x = \omega^2 \mu_0 \varepsilon_0 B^{3D} x \]  

(21)
Appendix
Target frequency in CEM3D

where

\[ A_{ij}^{3D} = \int \int \int_{\Omega} \frac{1}{\mu_r} \nabla \times \vec{w}_i \cdot \nabla \times \vec{w}_j d\Omega \]  \hspace{1cm} (22)

\[ B_{ij}^{3D} = \int \int \int_{\Omega} \varepsilon_r \vec{w}_i \cdot \vec{w}_j d\Omega \]  \hspace{1cm} (23)

\[ C_{ij}^{3D} = \int \int \int_{\Omega} \sigma \vec{w}_i \cdot \vec{w}_j d\Omega \]  \hspace{1cm} (24)
Appendix
Target frequency in CEM3D

Equation 21 can be rewritten as follows

\[(A^{3D} + j\omega_0C^{3D})x = (\frac{\omega}{C_0})^2 B^{3D} x\]  \hspace{1cm} (25)

\[s(A^{3D} + j\omega_0C^{3D})x = s^2(\frac{\omega}{C_0})^2 \frac{B^{3D}}{s} x\]  \hspace{1cm} (26)

\[A^{CEM3D} x = \lambda B^{CEM3D} x\]  \hspace{1cm} (27)

where

\[s = \frac{C_0}{2\pi f_T}\]  \hspace{1cm} (28)