

# 二次元DBS

---

## 概要

藤澤先生より頂いたWFMDBS-ver3-PCMozaicの流れを汲む。 不要な部分(wfmなど)をバッサリとカットしてDBSに特化している。

## 更新履歴

- ver 1 (2021/4/14)  
動作確認および完成
- ver 2 (5/8)  
マルチプロセス機能の削除。  
PixelReductionを $y=z$ の対称条件以外でも使用可能に
- ver 3 (7/16) 複素振幅係数の計算機能追加, 位相が計算可能に  
(位相計算の出力はprobaability.cで設定しているが、  
現在PowerSplitter設計時のみの出力)
- ver 3.1 (10/20) InputField読み込み時に、振幅の実部最大値は正の数になるように修正(port.c)
- ver 4.0 (2022/05/25) 理想界との重なり積分を計算する際に、複素共役を取る方が反対になっていたの  
で修正 修正等価屈折率について、3材料の割り当てを可能に 規格化について、3次元と揃える
- ver 4.1 (2022/06/15) fieldX, Yについて、プログラム順を入れ替えて、余計なファイル開閉削減による実行時間短縮

## 使用法

```
.../optfem -mz -f -DBS0 -WC test
```

```
~/Solver/2D-DBS-PCMozaic/optfem -mz -f -DBS0 -PS test
```

- -mz, -f, -DBS0  
前から順に、モザイク構造を使用して / 入射界はファイルで準備し、 / DBSを計算
- "test"  
test.mshやtest.preなどで使用するファイル名
- 最適構造(GuessedMatrix.csv)の計算をしたい  
.../optfem -mz -f **-fmz-2** test  
などとして実行する
- -WC, -PSなど  
設計するデバイスによってFOMを変更する必要があるので、ここで指定している。  
WaveguideCrossing:WC, PowerSplitter:PSなど

## 用意するもの

太字はDBSならではの部分

- Inputディレクトリ  
InputField-[id]-[wavelength] cross sectionの小さい方から順にidを振り、入射界を用意
- **mosaic.pre**

```

leftup(x,y) 2 10.2      // モザイク領域の左上(xが最も小さく、yが最も大きいところ)の座標
Nx,Ny 36 36            // 各方向のピクセル数
dx,dy 0.2 0.2          // 各方向のピクセルサイズ(単位:μm)
symflag 1               // 対称性を指定(最上位bit [y=z flag][zflag][yflag] 最下
位bit)
matIDofmozaic 3         // 可変とする材料番号(以下3つは番号が0始まりなので、GiDと違う)
coreIDofmozaic 0        // コアに割り当てた時の材料番号
cladIDofmozaic 2        // クラッドに割り当てた時の材料番号
gamma 0.5               // PowerSplitterの分岐比などで使用
DBSitr 10               // DBSの最大周回数(terminationによって、ここまで回らないかも)
termination 0.005       // 収束判定条件。各iterationにおけるfitnessの変化量を比較
PixelReduction 1        // モザイクパターンcsvファイルにて、判定しないピクセルは-1で明記してい
る。(0: off, otherwise: on)

```

- test.pre

```

wavelength information
2.10 2.10 0.01
modeID 2
Power = 1.0
the number of material = 6
use modifiedEIM = 0
re_index re_mu n2 n_sat
    2.551    1.0000000000 0.000000 0.000000
    2.551    1.0000000000 0.000000 0.000000
    1.235    1.0000000000 0.000000 0.000000
    2.551    1.0000000000 0.000000 0.000000
    1.235    1.0000000000 0.000000 0.000000
    2.551    1.0000000000 0.000000 0.000000
Field related parameters
div_x = 500
div_y = 500
xx cross section 0
division_y = 2000
center_x,inputornot
yy cross section 3
division_x = 1000
center_y,inputornot
1.0 1
10.2 0
10.2 0
num of light input = 1
pml_param = 2, 1.0e-6
pml = 2, 2
1.0    -40.0    40.0    -1.0
10.2   -40.0    40.0     1.0
1.0    -40.0    40.0    -1.0
12.2   -40.0    40.0     1.0

```

- test.msh  
解析対象のメッシュファイル
- index.pre  
test.preでuse modifiedEIMを有効にした場合のみ必要。  
波長ごとに使用する屈折率を設定する。(入射モードごとに変える場合も)

```
refractive index information (list order 1:core 2:clad)
wavelength 2.08
    2.561 1.233
wavelength 2.10
    2.551 1.235
wavelength 2.12
    2.541 1.237
```

- **MosaicPattern.csv**  
材料の初期パターン。  
一行内に全てずらっと書いてもいいが、可読性から適宜改行を入れ  
y座標と行番号を対応させている。  
mosaic.preでPixelReductionを1にしているときは、  
-1のピクセルを判定対象としない。

## 出力ファイル

- Field\_[in\_portID]-[wavelength].slv  
フィールド分布
- fieldY\_[in\_portID]-[out\_portID]-[wavelength]  
ポートにおける界分布。test.preのdivision\_yで指定された分割数で出力される。
- **fitness.data**  
各ピクセル判定におけるfitness関数の値を出力する。  
書式

```
[no]: [calculated],[optimum],[changed] / [values:複数波長の計算の時などは、計算に  
用いた特徴量をそのまま出力]
```

- **GuessedMatrix.csv**  
GuessedMatrix.csvは最終的に得る構造,  
GuessedMatrix-[itr-1].csvは、各iterationの終了後(一通りピクセルを判定した後)における最適構造
- Overlap  
最適構造における透過率を出力
- **pattern.data**  
採択された変更を、順次追記していく。

```
[no], [fitness]:[ピクセル状態]
```

これを使用することで、状態変更の推移を簡単に出力可能となる。(追計算もしやすい)

- portfield  
本来は正規化した入射界を出力しておくのだが、現在設定しておらず
- **sequence.data**  
ピクセルの選択順を出力しておく。  
仮に計算途中でサーバーが落ちたときなども、再開が容易に。  
また、ピクセル選択順が本当にランダムかの確認等も
- XX, YY  
フィールド分布描画の際に必要となる。  
「.slvファイルで電磁界値を出力している座標」を出力している。
- **YY-Mosaic, ZZ-Mosaic**  
モザイクのピクセルについて、中心座標を出力している。

## 注意点

- ~~同時に回すプロセスの数は波長の数と同じであるとみなして設計している。~~  
2次元設計においては省メモリ・省プロセスの方が優先度が高いとして、  
複数波長最適化はシングルプロセスで行う。(5/8)
- ~~並列化も、波長に関してのみを考えた設計となっており、  
導波路構造は全く同じであるとする。~~  
並列計算機能は削除 (5/8)
- 諸解析結果の出力は、必要であれば適宜挿入するものとしておき、  
DBSの反復回数などの細かい情報は区別しない。  
**ファイル出力するのは最終周の最適構造のみ(.slv)**
- DBSの収束条件を設定できるようにしたほうがよさそう。(実装済み)  
最大周回数とfitnessの変化量の両方を対象
- PixelReductionの判定は、0かそれ以外

## 将来的に実装を考えている機能