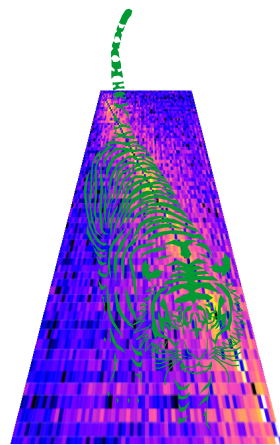


SMatrAn (ver. 0.150117) のてびき

臼杵 達哉

2015 年 4 月 2 日



SMatrAn
Scattering Matrix Analyzer

SMatrAn は波動方程式から導出される散乱行列の数値シミュレータです。散乱行列を出力出来るシミュレータは他にもありますが、他の形式の計算結果から一部の行列要素に変換する方法が一般的です。一方で、構造物によって散乱された散乱波の定量的評価は、散乱行列から直接得られます。反射波や透過波の振幅や位相情報は散乱行列に全て含まれているからです。SMatrAn の機能は散乱行列全要素の高精度計算に特化しています。したがって、技術者や科学者の方々に有益な情報を SMatrAn は提供出来ます。

今回、電磁波の散乱行列を求める SMatrAn のソースファイルを公開しました。どなたでもダウンロードして使用することが出来ます。但し、利用者が一度はソースファイルを読んでもらえるように、実行ファイルのコンパイルは各自でお願いしています。Linux は勿論ですが、Windows においても MinGW を用いれば手軽にフリーのコンパイラをインストール出来ます。(Windows 上で SMatrAn をインストールしたり実行するには Windows コマンド シェルを用います。MSYS は不要です。) WEB 上には、コンパイルを実行するスクリプトと 3 次元光学構造の例題を用意しています。ソースファイルやその他のファイルを含んだディレクトリ全体をダウンロードすれば、迅速にシミュレーション環境を構築できます。

SMatrAn の実行はバッチ形式で行います。入力ファイル生成の GUI を用意していないので、入力ファイルを直接扱う必要があります。このマニュアルは SMatrAn の簡単な紹介の後、先ず実行環境構築の代表的な手順について解説します。次に例題の計算を基にして、入力ファイルの編集や出力ファイルの解析方法を実際の手順を紹介しながら説明します。

このマニュアルを読めば、一通り SMatrAn を使えるようになります。光学構造の複雑な散乱も、散乱行列を通して詳細な解析が行えるでしょう。個人で開発しているので限りがありますが、コードや計算方法の改良は出来るだけ対応いたします。質問やコメントがございましたら、お気軽に著者に御連絡下さい。

目次

1	SMatrAn とは	4
1.1	SMatrAn 開発の背景	4
1.2	SMatrAn の利用者と計算対象	5
1.3	使用プログラム言語と環境	6
2	SMatrAn の構成	8
2.1	Preparation group	8
2.2	Simulation group	9
2.3	入力書式	9
3	SMatrAn の構築	11
3.1	実行ファイルのコンパイル	11
3.2	例題の実行	12
4	Preparation group 内の数値処理の流れ	15
4.1	Mesh3D の操作	15
4.2	YeeCells operation: Yee's cell calculation	16
4.3	Med3D operation	17
4.3.1	Slicing operation	19
5	Simulation group 内の数値処理の流れ	20
5.1	Mode calculation	20
5.2	Stable transfer matrix method	21
6	Development schedule and history	22
7	LICENSE AGREEMENT	23

1 SMatrAn とは

SMatrAn は波の散乱を記述する散乱行列を数値計算します。散乱行列に特化した数値計算法の名称に適切なものが無かったので、本文では ‘Scattering Matrix Analyzer’ または ‘Scattering Matrix Analysis’ の呼称として ‘SMatrAn’ を用いることにしました。図 1.1 は WEB や発表資料で用いる



図 1.1: Logo of SMatrAn.

logo です。デバイス設計者をはじめとする一般の利用者にこの計算法を知ってもらうため、呼称や logo を考えました。これらを規定するつもりはないので ‘SMATRAN’ と綴っていただいても構いません。また SMatrAn の発音は決めていませんが、*sumatran* を連想して名づけました。

本章では開発の背景から開発環境まで説明します。第 2 章は、公開された SMatrAn に関する実行ファイルの紹介です。第 3 章は、動作環境の構築と例題の実行方法の解説です。第 4 章と第 5 章は例題を基にした各種パラメータの設定方法や計算出力の詳細、第 6 章は今後の予定について触れています。時間のない方は第 3 章から読み始めて頂いても構いませんが、第 7 章の “LICENSE AGREEMENT” は必ず目を通して下さい。質問やコメントがございましたら、お気軽に著者へ御連絡下さい。

1.1 SMatrAn 開発の背景

散乱行列要素の精密な解析は、不均一な媒質中を伝播する波と、媒質の両端で厳密に定義された導波路モードとの間に接続条件を課して行います。一例として、形状の異なる導波路を結合する 3 次元構造物 (図 1.2) を御覧下さい。‘Bottom waveguide’ (図では左側) から入射した波が、中央の ‘Joint structure’ を介して、‘Bottom waveguide’ へ反射する波と ‘Top waveguide’ (図では右側) へ透過する波に散乱されます。散乱行列の各要素 (複素数) は、散乱されたモードに関する入射モードで規格化された流量の平方根とその位相に対応します。導波路モードとして図 1.2 で描かれた波

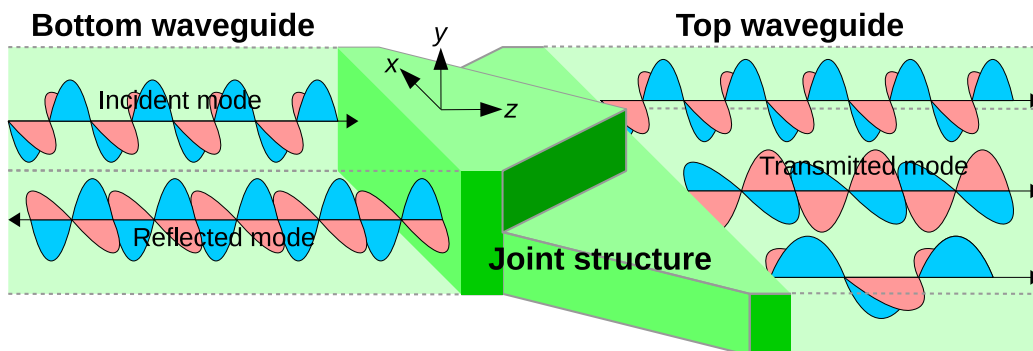


図 1.2: Joint structure between two waveguides is shown as a 3D example. The z-axis is oriented from the bottom to the top.

はかなり単純化されていますが、実際の数値計算ではエバネッセントモードも含めたあらゆる波を考慮します。正確な散乱行列の定義と計算法の詳細は文献 [1, 2, 3] を御精読願います。

SMatrAn は波の伝播を規定する波動方程式に特別な制限を必要としません。今までに Shrödinger equation や Maxwell equation に基づく散乱行列解析に SMatrAn を適用しました。SMatrAn の基になる手法が最初に使用されたのは、メゾスコピック領域における 2 次元電子系電気伝導率の計算です [1]。それまでも量子輸送現象の数値計算には matrix method [4] や recursive Green-function method [5] 等がありました。前者は量子井戸の解析に適した安定的な計算方法です。但し散乱体内部で連続的に固有値問題を解く必要があり、複雑な構造では各ステップの固有ベクトルの誤差も累積します。後者は 2 次元電子系、特にアンダーソン局在のような複雑な散乱 [5] を解析するためによく用いられました。途中で固有値計算する必要はありませんが、条件によって数値計算が不安定になります。文献 [1] の手法は両者の問題点を解決し安定的に精度良く計算できるため、今でもグラフェンをはじめとするメゾスコピック系の量子輸送問題 [6, 7, 8] に用いられています。

一般公開した **SMatrAn** は、フルウェーブ 3 次元電磁界 (EM) シミュレータです。EM シミュレータに絞った理由は、SMatrAn のもつ安定性と解析精度が、シリコンフォトリソグラフィを始めとする導波光学向けや高周波デバイスの設計検証に広く役立つからです。様々な方からのフィードバックでシミュレータの信頼性が向上することも一般公開によって期待されます。

1.2 SMatrAn の利用者と計算対象

公開した SMatrAn の利用者として、既に何らかの電磁界シミュレータを使用している技術者や科学者を想定しています。(特に仕事として) 数値シミュレーションを初めて行うのなら、その分野で広く普及しているシミュレータを選んだ方が良いでしょう。高額な市販品には、充実したサポートや CAD 連携等の利点があります。それでも、汎用シミュレータでは取り扱うのが難しい解析が存在します。

導波光学系の反射率、モード間遷移、可逆性、或いは直交性を周波数領域で“精密”に調べる必要に迫られた時こそ、SMatrAn を利用すべきです (図 1.3)。多くの導波光学系や伝送線路がこれら

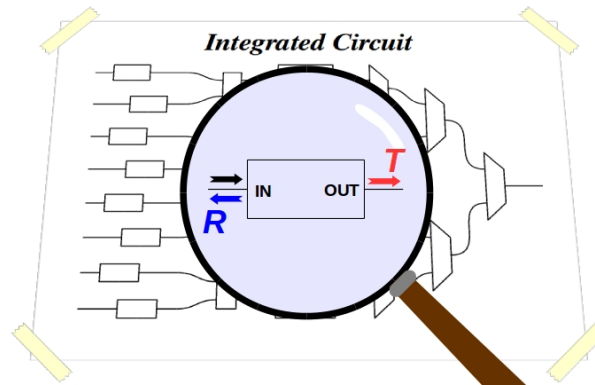


図 1.3: SMatrAn is just like a magnifying glass for designing devices.

の特性を指標としているので、特性評価に SMatrAn が出力する散乱行列 (S-parameter の名称の方が馴染みがあるかもしれませんが) が直接役立ちます。

SMatrAn は、波動方程式の離散化と数値計算で生ずる数値誤差以外、何も近似をしません。離散化は流量の保存則を破らないように行いますので、波の伝播に関する性質を守ります。また、数値誤差が大きくなりえない安定的な陰解法を採用している上、誤差が生じやすい固有ベクトル計算では必要に応じて計算精度を向上させるライブラリを用意しています。これらの対策によって、散乱行列に関する残差が 10^{-5} 未満 (通常は 10^{-10} 未満) を満足する様に SMatrAn は設計されています。

必要に応じて高速化も進めますが、精度の向上や散乱の評価法に関わる改良を優先させますので、ご了承ください。数値計算の分野では、“速い”、“でかい”、“正しい”が正義だそうです [9]。この中

で、SMatrAn は散乱現象に関わる“正しい”を最優先しています [10]。高速化のために LAPACK の使用を一部試みっていますが、精度検証を優先させるため LAPACK や BLAS の適用は遅れがちです。SMatrAn の設計指針は、速度や規模の為に陽解法を使用する FDTD 法 (Finite-difference time-domain method) とは正反対であることに留意下さい。

SMatrAn 環境の構築と利用には、ソースコードからプログラムをコンパイルし、shell script を扱う技能 (大した水準ではありません) が必要です。SMatrAn の使用によって発生したあらゆる損害について著者は責任を負わないものとします。最 7 章の“LICENSE AGREEMENT”に記載したライセンスを御一読下さい。

1.3 使用プログラム言語と環境

SMatrAn は C99 で書かれています。文献 [1] を基にして電磁界解析向けに拡張を考えたのが 2005 年、導波路モード計算を始めたのが 2010 年、そして電磁界解析で散乱行列を初めて扱った [2, 3] のが 2012 年の夏でした。これらのプログラムに最小限の修正と拡充を (2014 年 4 月から) 施したのが、SMatrAn の first version です。

単精度・倍精度・拡張倍精度の数学関数とそれぞれの複素関数に関して、C99 の規格は C++98 や Fortran と比べ格段に優れています [11]。<tgmath.h>も計算プログラムを書くときに便利です。但し、規格に対応しているコンパイラが少ないという欠点もあります。当時から GNU Compiler Collection (GCC) を利用していましたので、C99 を選択したのは妥当でした。しかし月日は流れ、C++11 には C99 の数学関数を取り入れられ、<ctgmath>も実装されました。他の実装は C99 より C++11 の方が進んでいる上 [12]、Windows の Visual C++ は C++11 に対応しているので、次のプログラム改訂では部分的に C++11 を使用するかもしれません (図 1.4)。

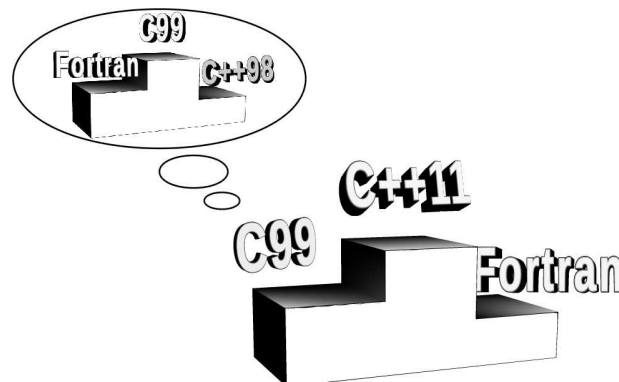


図 1.4: The standard of mathematical functions in C++11 is now equal to or better than that in C99.

OS は Ubuntu14.04 LTS の 32 bit を SMatrAn の開発環境として使用しています。SMatrAn の検証を Ubuntu14.04 LTS 64 bit と Windows 7 32 bit 及び 64 bitで行っています。コンパイラは GCC version 4.8.2 以降と Clang (based on LLVM 3.4 以降) を用います。Windows での GCC は MinGW (GCC version 4.8.1-4 以降) です。つまり SMatrAn は OS とコンパイラの 8 種類の組合せで検証しています。これらのコンパイラは GPL や University of Illinois/NCSA License で広く普及しています。

SMatrAn の開発に利用しているハードウェアは、主に表 1.1 の 2 種類です。ソースコードにハード依存 (並列処理等) する箇所はありませんので他の CPU でも動作するはずですが、試してはいません。ただし、固有ベクトル計算で拡張倍精度ライブラリを使用する場合、Intel の 80-bit 浮動小数点演算の方が (他の倍精度計算の影響を含めても) 精度が 1 桁程度は高くなります。開発環境 (32 bit) のハードは現在ではかなり低い性能ですが、試用なら十分な計算速度 (64 bit 機の 1/5 程度) です。

表 1.1: Two hardwares for this manual.

Architecture	CPU	Clock	Memory	Purpose
32 bit	U2300	1.20 GHz×2	3.9 GiB	Developement tool
64 bit	i7-4770	3.40 GHz×8	7.7 GiB	Verification tool

次章以降で SMatrAn の詳細を説明します。

[\[Go to table of contents.\]](#) [\[Go to home.\]](#)

2 SMatrAnの構成

本章ではSMatrAnを構成する実行ファイルを紹介します。図2.1に示すように9つの実行ファイルがありますが、3つの実行ファイル群にまとめられます。まず波動方程式を形作る3次元パラメータ

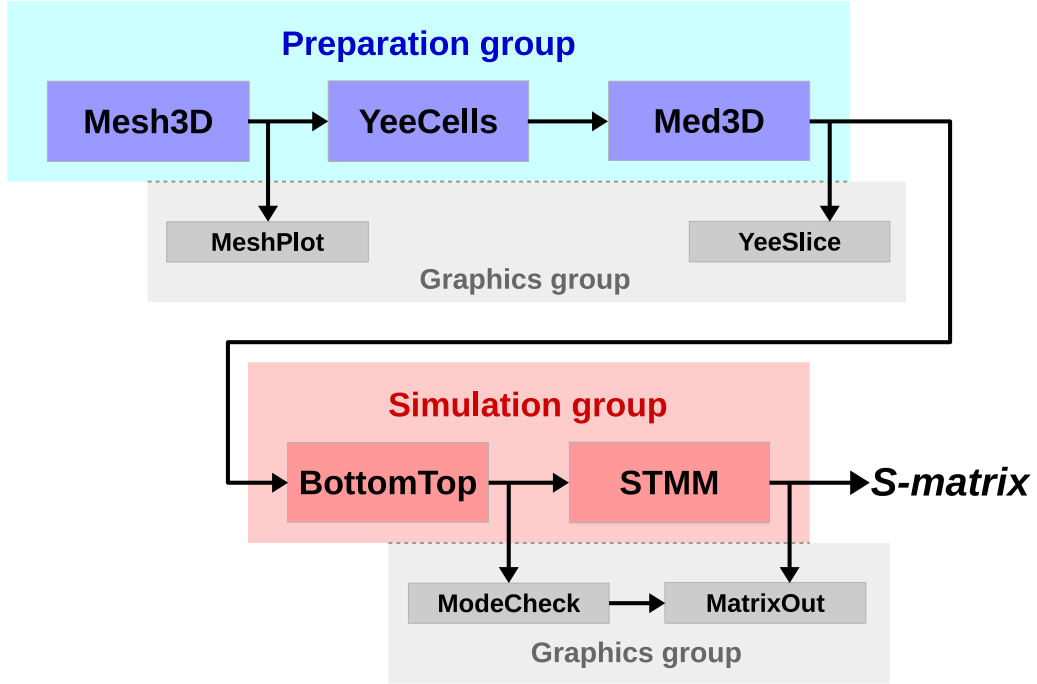


図 2.1: Simulation flow from 3D-meshing to S-matrix.

を生成を担う preparation group と呼ばれる実行ファイル群があります。図2.1では、preparation group を青色で示します。次に波動方程式を計算して散乱行列を導く処理群を simulation group と呼び赤色で示します。そしてそれぞれの実行ファイルが出力する生成ファイルを検証するために、灰色で示した実行ファイル群 (graphics group) を用意しています。

2.1 Preparation group

パラメータを生成を担う preparation group は3つの実行ファイルで構成されます。

1. Exec. file name: Mesh3D
 - This file generates nonuniform mesh for Yee's lattice.
2. Exec. file name: YeeCells
 - This file calculates optical structure in each Yee's cell.
3. Exec. file name: Med3D
 - This file calculates electric permittivity and magnetic permeability in each Yee's cell.

3つの実行ファイルを連続的に実行すれば、simulation groupに必要な3次元データは全て揃います。但し、3次元構造を適切に設定できているか確認する場合、生成された数値データだけでは判断し難いです。そこで数値データを可視化するため、graphics groupの2つの実行ファイルを使用します。

1. Exec. file name: MeshPlot
 - This file for checking Yee's lattice.
2. Exec. file name: YeeSlice
 - Slicing operation for checking 3D permittivity distribution.

Graphics groupを用いて、cellを規定するmeshとcellに与える誘電率の空間分布が確認できたら、simulation groupを扱います。

2.2 Simulation group

波動方程式を計算するsimulation groupは2つの実行ファイルで構成されます。

1. Exec. file name: BottomTop
 - This file calculates waveguide/evanescent-modes for both the bottom and top waveguides.
2. Exec. file name: STMM
 - This file calculates S-matrix by the transfer matrix method.

2つの実行ファイルを実行すれば、散乱行列を求めることが出来ます。散乱行列のユニタリ性などの特質は、STMMが出力します。各導波モードの分布や行列要素の残差分布を把握するには、graphics groupの2つの可視化ツールを使用します。

1. Exec. file name: ModeCheck
 - This file for checking orthogonality and generating GNUplot data-files for the waveguide/evanescent modes.
2. Exec. file name: MatrixOut
 - This file maps matrix elements.

紹介した9つの実行プログラムを実行する際、使用者が用意する制御ファイルとその他の実行プログラムが出力するデータファイルのどちらか或いは両方を入力します。どちらの入力ファイルも可読性のためにASCII-textを用いています。制御ファイルの準備には、書式についての理解が必要です。

2.3 入力書式

SMatrAnには幾つかの制御ファイルが必要です。制御ファイルは利用者自ら作成して下さい。1行あたりの文字の長さは空白も含め1024文字未満で、マルチバイト文字は非対応です。

実行プログラムは何らかの文字に続く '=' の後の数字 (又は文字) を読み込みます (空白は無視)。幾つかの例を示します:

True L = 22, K_u = 4

True L=22K_u=4

False =22,K_u=4

物理単位が必要な場合、次の単位が使用可能です: 'deg' as °, 'rad', 'km', 'm', 'cm', 'mm', 'micron', 'um', 'nm', 'K', 'degC' as °C and 'degF' as °F. 実行プログラムは、次の数量と区別するため単位の後空白を必要とします。

True x_min = -1.5 micron x_max = 1.5 um

True x_min = -1.5micron, x_max=1.5 um

False x_min = -1.5micron,x_max= 1.5 um

以上が SMatrAn を構築するのに必要な事項です。

次章では例題を用いながら SMatrAn の環境構築を紹介します。

[[Go to table of contents.](#)] [[Go to home.](#)]

3 SMatrAn の構築

本章で取り扱う SMatrAn の例題を紹介します。例題を含む SMatrAn の構築及び動作によって起こる問題に関して、**くどいようですが**著者は責任を負いません。マニュアルの pdf 形式を除く、全てのファイルは ASCII text です。利用者はファイルと第 7 章の “LICENSE AGREEMENT” を読み内容を十分に理解した後、自己責任でご利用下さい。“**使用プログラム言語と環境**” で述べたように、Ubuntu と GCC の組合せで開発しています。他の組合せでご利用の方は、必要に応じて shell script を書き換えて下さい。以下では説明しませんが、Clang の shell script や Windows 向けの batch file も用意しています。Batch file はマウスでクリックすれば動作します。

3.1 実行ファイルのコンパイル

例題では図 3.1 に示したディレクトリ構造を用います。全体の容量は、最初マニュアル類を除いて 397 kB ですが、シミュレーション終了時には 81.7 MB まで増加します。(これ以外のディレクトリ

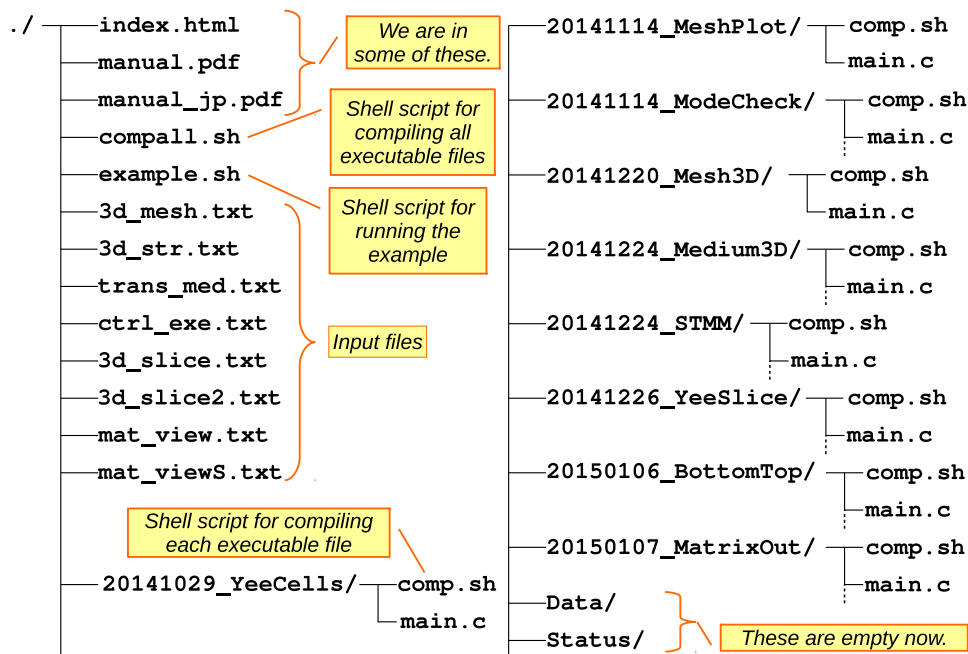


図 3.1: An example of initial files and directories. ‘./Data/’ and ‘./Status/’ are empty before running SMatrAn.

構造で動かす場合は、shell script と幾つかの制御ファイルを変更する必要があります。) 前節で述べたように、SMatrAn は 9 つの実行プログラムで構成されています。本マニュアルがあるディレクトリ (図 3.1) には実行プログラムに対応したディレクトリが 9 つあります。

図 3.1 に示したファイルは全てダウンロード可能です。それぞれは小さなプログラムなので、コンパイルに ‘make’ は用いず shell script で gcc コマンドを実行します。(Windows 環境で batch file を利用する場合、MSYS 無しで MinGW 或いは MingGW+Clang のみの環境で構いません。) 各プログラムのディレクトリに更新期日を明示していますので、ダウンロード時の time-stamp 管理

を気にする必要はありません。例えばディレクトリ ./20141220_Mesh3D の下には C99 ソースファイルとヘッダファイルの他に comp.sh があります。comp.sh の内容は

```
#!/bin/sh
gcc -Wall -O3 *.c -std=C11 -lm -o ../Mesh3D 2> error_gcc.txt && echo '
    compiled Mesh3D'
```

必要に応じて shebang や compile option は変更して下さい。ターミナルを立ち上げ ./20141220_Mesh3D に移動します。shell を実行する前に必ず comp.sh の内容を確認して下さい。

```
~/SMatrAn/20141220_Mesh3D$ sh comp.sh
```

(上記は、ホームディレクトリ SMatrAn の下にディレクトリ SourceFiles を作成し、その中に図 3.1 のディレクトリが構築されている場合の表示です。)

コンパイルに成功したら、./20141220_Mesh3D の下に空の error_gcc.txt が生成されます。もし空でない場合は、error/warning message に従って compile option や source file を修正して下さい。他の実行ファイルも同じ方法で生成できますが、最初に 9 つの全実行ファイルをコンパイルするなら ./compall.sh が便利です。BottomTop は高速動作のため LAPACK の 'dgeev' を使用することが出来ます。LAPACK 環境を構築している方は、./20150106_BottomTop/下のファイル名末尾に .lapack があるファイルを、末尾に .lapack がないファイルと入れ替えた上でコンパイルして下さい。今後、他の実行ファイルでも対応は遅れがちですが LAPACK/BLAS を利用する可能性が有ります。高速化が必要な場合は LAPACK/BLAS のインストールをお勧めします。

3.2 例題の実行

実行ファイルを全てコンパイルしたら、例題を計算することができます。コンパイルに使用したディレクトリは今後使いませんので、不必要なら削除しても構いません。確認のため、計算直前のファイルとディレクトリを図 3.2 に示します。

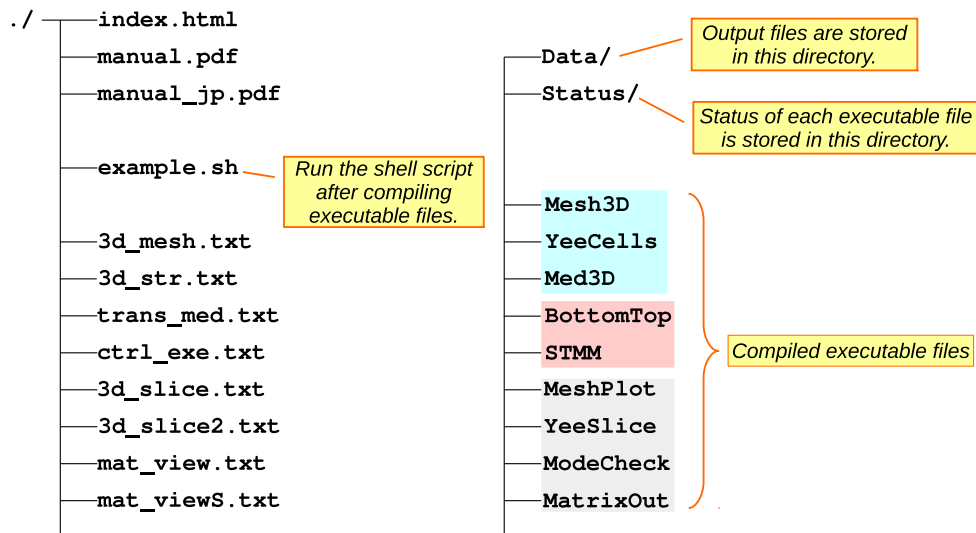


図 3.2: Files and directories after compiling all executable files and removing unnecessary directories.

9 つの実行ファイルを順に実行するため、シェルスクリプト example.sh を用います。(Windows では ./example.sh の代わりにバッチファイル ./example.bat を残して下さい。) ./example.sh の内容を下記に示します。

```
#!/bin/sh
./Mesh3D 3d_mesh.txt ./Data/3d_mesh.dat 2> ./Status/Mesh3D.log && \
echo 'Mesh3D finished' && \
./MeshPlot ./Data/3d_mesh.dat 2> ./Status/MeshPlot.log && \
echo 'MeshPlot finished' && \
./YeeCells ./Data/3d_mesh.dat 3d_str.txt ctrl_exe.txt 2> ./Status/
  YeeCells.log && \
echo 'YeeCells finished' && \
./Med3D trans_med.txt ctrl_exe.txt 2> ./Status/Med3D.log && \
echo 'Med3D finished' && \
./YeeSlice ./Data/3d_mesh.dat 3d_slice.txt 2> ./Status/YeeSlice.log && \
echo 'YeeSlice finished' && \
./YeeSlice ./Data/3d_mesh.dat 3d_slice2.txt 2> ./Status/YeeSlice.log &&
\
echo 'YeeSlice finished' && \
./BottomTop ./Data/3d_mesh.dat ctrl_exe.txt 2> ./Status/BottomTop.log &&
\
echo 'BottomTop finished' && \
./ModeCheck ./Data/3d_mesh.dat ctrl_exe.txt 2> ./Status/ModeCheck.log &&
\
echo 'ModeCheck finished' && \
./MatrixOut ./Data/joint_Bottom_check.dat mat_view.txt 2> ./Status/
  MatrixOut_B.log && \
echo 'MatrixOut finished' && \
./MatrixOut ./Data/joint_Top_check.dat mat_view.txt 2> ./Status/
  MatrixOut_T.log && \
echo 'MatrixOut finished' && \
./STMM ./Data/3d_mesh.dat ctrl_exe.txt 2> ./Status/STMM.log && \
echo 'STMM finished'
./MatrixOut ./Data/joint_final.dat mat_views.txt 2> ./Status/MatrixOut_S
  .log && \
echo 'MatrixOut finished' && \
```

上記の内容を確認した上で例題の計算に進みます。

```
~/SMatrAn$ sh example.sh
```

計算が進む毎に、ターミナルに終了メッセージが表示されます。全て終了すると、下記の様にコマンド入力待ちになります。(Windows ではコンソールが閉じる。)

```
~/SMatrAn$ sh example.sh
Mesh3D finished
MeshPlot finished
YeeCells finished
Med3D finished
YeeSlice finished
YeeSlice finished
BottomTop finished
ModeCheck finished
MatrixOut finished
MatrixOut finished
STMM finished
MatrixOut finished
~/SMatrAn$
```

表 3.1: Compiler dependence of numerical calculation on several OSes. $\|\mathbf{S}^\dagger \mathbf{S} - \mathbf{I}\|_{\max}$ shows an index as precision of S-matrix. Note that $\|\mathbf{A}\|_{\max}$ is the max norm defined as $\|\mathbf{A}\|_{\max} \triangleq \max |a_{ij}|$. Specifications of 64 bit and 32 bit hardwares are shown in Table 1.1.

OS	item	GCC	Clang
Ubuntu 64 bit	$\ \mathbf{S}^\dagger \mathbf{S} - \mathbf{I}\ _{\max}$	1.88642×10^{-13}	1.74536×10^{-13}
	STMM time (s)	37	24
	Total time (s)	80	66
Windows 7 64 bit	$\ \mathbf{S}^\dagger \mathbf{S} - \mathbf{I}\ _{\max}$	3.16025×10^{-13}	1.89627×10^{-13}
	STMM time (s)	184	29
	Total time (s)	256	83
Ubuntu 32 bit	$\ \mathbf{S}^\dagger \mathbf{S} - \mathbf{I}\ _{\max}$	8.48794×10^{-14}	1.57441×10^{-13}
	STMM time (s)	636	140
	Total time (s)	939	342
Windows 7 32 bit	$\ \mathbf{S}^\dagger \mathbf{S} - \mathbf{I}\ _{\max}$	3.19174×10^{-13}	1.99567×10^{-13}
	STMM time (s)	670	176
	Total time (s)	994	447

参考までに、著者の環境における例題の精度と計算時間を表 3.1 に示します。この組合せの範囲内で Windows 7 より Ubuntu の方が、そして GCC より Clang の方が計算速度は速い傾向があります。計算精度は環境の影響が大きく出ないようにプログラムを改良していますが、Clang の方が OS の影響が少ない様です。従いまして、どちらの OS を使うにしても現時点でのコンパイラは Clang の使用をお薦めします。

また ./Data/ 内の出力ファイルは 81 MB に達します。ここまで完了すれば SMatrAn を使用する環境が整います。次章以降でそれぞれの実行ファイル群に関する入出力ファイルの流れを説明します。入力パラメータの設定方法を理解すると、他の光学構造でも散乱行列を計算できるようになります。

[\[Go to table of contents.\]](#) [\[Go to home.\]](#)

4 Preparation group 内の数値処理の流れ

本章では preparation group (図 2.1) の入出力ファイルについて例題 (図 3.2) を用いながら説明します。図 4.1 に入出力ファイルの流れを示します。

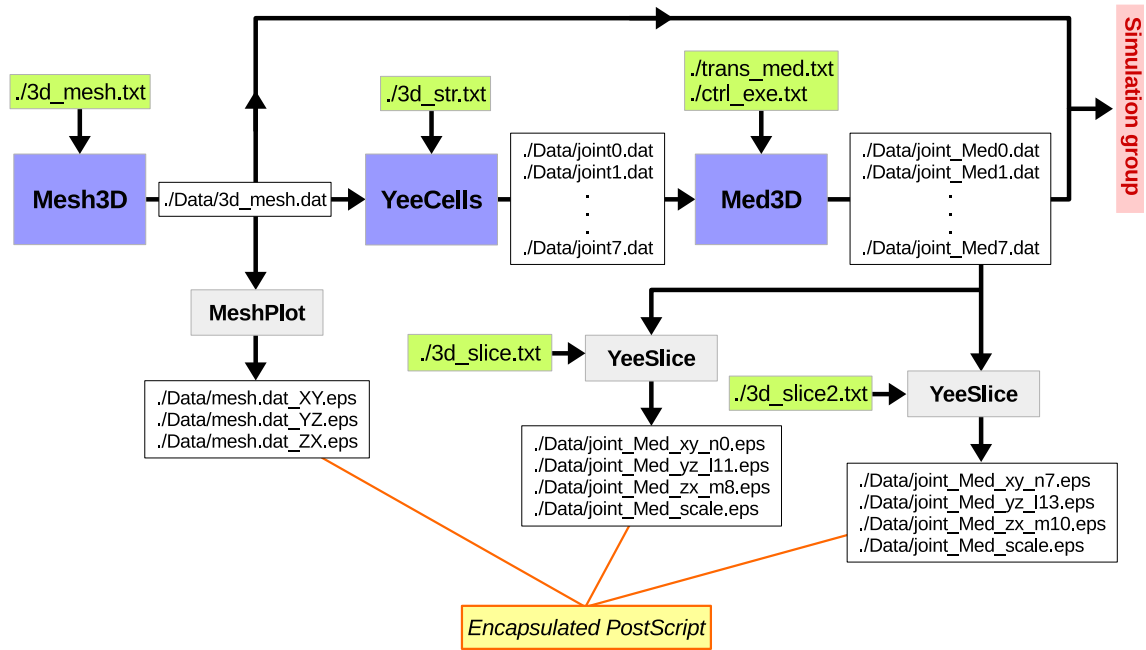


図 4.1: Data flows from Mesh3D to Med3D in the preparation group. White boxes with black frame show output-file names from executable files. All outputs are ASCII file, and outputs from MeshPlot and YeeSlice in the graphics group are formatted by the encapsulated postscript. Yellow-green boxes mean control files that users write.

4.1 Mesh3D の操作

Mesh3D は 1 つの input file で不均一メッシュデータを作成します。Numerical simulator consists of four programs and one checking program.

1. Nonuniform mesh generation
2. Yee's lattice generation
 - a) Slicing operation for checking generated Yee's lattice

3. Mode calculation
4. Transfer matrix method

We shows a program of generating mesh[?].

- program name: `Mesh3D.exe`
 - source file archive: `Mesh3D_20120723.zip`
 - example batch file: `Mesh3D.bat`
 - example input file: `Frame_3um4um28p4um.txt`
This subsection gives us details of the file.
 - example output file: `Frame_3um4um28p4um.dat`
This generated file is used by the following programs.
 - example error file: `error_Frame_3um4um28p4um.dat`
This file has some message if the program abnormally run.

The input file consists of 6-lines.

Input file to Mesh3D `./3d_mesh.txt`

1-st `L = 22, K_u = 4`

Positive integer L means max u for u -coordinate. Nonnegative integer K_u is used in the function F of Eq. (3) in Appendix I[?]. When we generate uniform mesh, we set $K_u = 0$.

2-nd `M = 18, K_v = 3`

Positive integer M is max v for v -coordinate. Nonnegative integer K_v is also used in the function F .

3-rd `N = 500, K_w = 0`

Positive integer N is max w for w -coordinate. This example shows uniform mesh, because $K_w = 0$.

4-th `x(0) = -1.5 micron, x(L) = 1.5 um, min dx/du = 80. nm`

Positive real $x(0)$, $x(L)$ and min dx/du require length unit.

5-th `y(0) = -2.0 um, y(M) = 2.0 micron, min dy/dv = 70. nm`

Positive real $y(0)$, $y(M)$ and min dy/dv require length unit.

6-th `z(0) = 0.0 micron, z(N) = 28.4 um, min dz/dw = 0.0 m`

Positive real $z(0)$, $z(N)$ and min dz/dw require length unit. The above min dz/dw is unused, because $K_w = 0$.

4.2 YeeCells operation: Yee's cell calculation

This subsection shows a program of generating Yee's lattice for a wave length.[?].

- program name: `YeeLattice.exe`
 - source file archive: `YeeLattice_20120708.zip`
 - example batch file: `YeeLattice.bat`

- example input file 1: `Frame_3um4um28p4um.dat`
This file is the output file from `Mesh3D.exe`.
- example input file 2: `DeviceStr.txt`
This subsection gives us details of the above file.
- example input file 3: `YeeCell.txt`
This subsection also gives us details of the above file.
- example output file: `YeeLatticeFile/Yeen.dat`
The program outputs $\varepsilon_\eta(l, m, n)$ and $\mu_\eta(l, m, n)$ when $\eta = u, v, w$, and $0 \leq n < N$.
- example error file: `logfile.txt`
For normal execution, the program outputs data of both the input file 1 and the input file 2 to this file. It has some message if the program abnormally run.

The input file 2 is described by commands for optical objects. Maximum line number should be less than 16384 in this file. Definition of each optical object starts at ‘begin’ command, and it finishes at ‘end’ command. Number of the objects should be less than 1024. Lower commands overwrites upper ones in the file. We have to set three commands between ‘begin’ and ‘end’, and we can also use two optional command.

4.3 Med3D operation

‘material’ This command has to be set with ‘name’ of material, ‘index’ for optical index, or ‘epsilon’ for relative permittivity.

Example1: material name SiO2

We can use ‘name’ for the wavelength λ : $1.25 \mu\text{m} \leq \lambda \leq 1.61 \mu\text{m}$. The ‘SiO2’ means SiO₂. Other material names are ‘Si’, ‘Al’, ‘Cu’, ‘Au’, ‘Ag’, ‘Pt’, ‘Ti’, ‘W’, ‘TiN’ and ‘Vacuum’. Initial permittivity in all space is set as ‘Vacuum.’

Example2: material index real= 1.5 imag= 0.

The optical index for a structure is given by complex number, and then we have to input real and imaginary parts.

Example3: material epsilon real= 3.6 imag= 0.

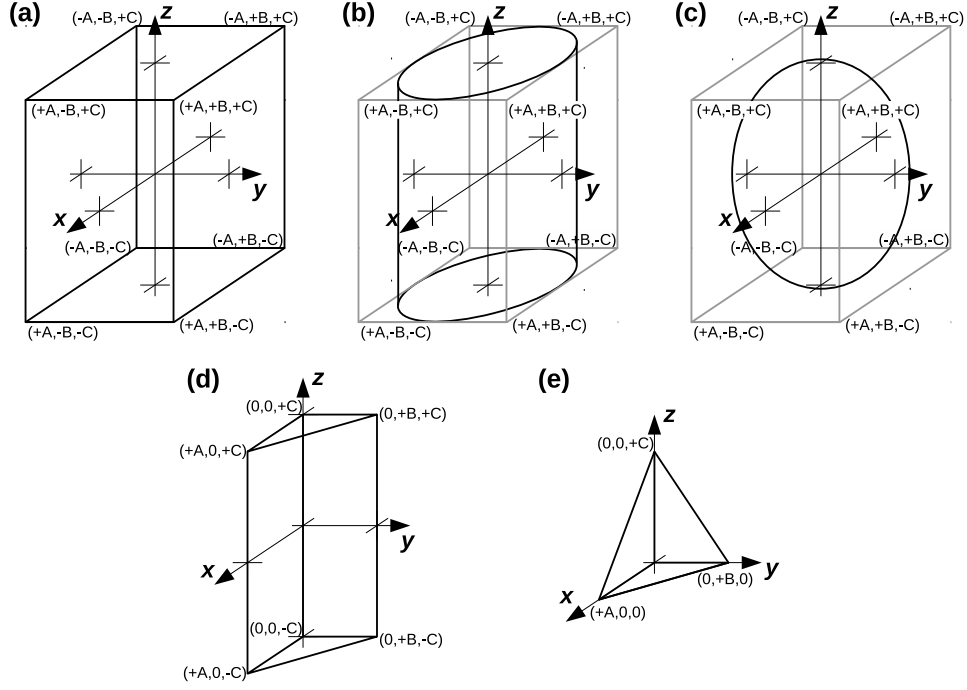
The relative permittivity is also complex number.

‘shape’ This command defines a shape of the optical structure with ‘block’, ‘cylinder’, ‘spheroid’, ‘prism’ or ‘pyramid’ for relative permittivity. We also have to set three positive real parameters A , B and C as shown in Fig. 4.2, and three real parameters θ_x , θ_y and θ_z for rotation angles. The first rotation is perpendicular to x -axis, the second one to y -axis, and the third one to z -axis.

Example: shape block A= 0.5 um, B= 200 nm, C= 13.5 micron, Theta_x= 0. deg, Theta_y= 0. rad, Theta_z = 0. deg

‘origin’ This command defines us a coordinate for the center of the optical structure.

Example: origin x= -3.0 um y= 0.0 um z= 14 micron



☒ 4.2: We can assign one of five shapes to an optical structure. (a)'block'. (b)'cylinder'. (c)'spheroid'. (d)'prism'. (e)'pyramid'.

'array' This is an optional command with a direction vector \mathbf{a} , integer I_{\min} and I_{\max} . It arrays an origin of each structure at $I\mathbf{a}$ for $I_{\min} \leq I \leq I_{\max}$. We can maximally use this command three times between 'begin' and 'end'. Maximal case is for setting 3D-grid.

Example: array a_x= 0.0 um, a_y= 0.0 um, a_z= 284 nm, I_min = 0, I_max = 89

'permeability' This is also an optional command for setting magnetic relative permeability μ . Initial value is set that $\mu = 1$.

Example: permeability real= 0.707 imag= 0.707

The program ignores any words except for the above 5 command in the input file 2.

The input file 3 consists of 3-lines, and an example of the file is shown as follows.

YeeCell.txt:

1-st Wavelength = 1.55 micron

Optical permittivity ε_η depends on the wavelength.

2-nd Data-file name = YeeLatticeFile/Yee

This file defines the first half of output file name. The program adds the last half 'n.dat' to it.

3-rd C_x= 2, C_y= 2, C_z= 0

C_η for $\eta = u, v, w$ is nonnegative integer. $2C_\eta + 1$ is point number for average along η -axis in each Yee's cell.

4.3.1 Slicing operation

This sub-subsection describe slicing operation on Yee's lattice to confirm the defined optical structures.

- program name: `YeeSlice.exe`

- source file archive: `YeeSlice_20120708.zip`

- example batch file: `YeeSlice.bat`

- example input file 1: `Frame_3um4um28p4um.dat`

- This file is the output file from `Mesh3D.exe`.

- example input file 2: `YeeSlice.txt`

- This sub-subsection gives us details of the above file.

- example input file 3: `YeeLatticeFile/Yeen.dat`

- This file is the output file from `YeeLattice.exe`. The n is nonnegative integer, and $0 \leq n < N$.

- example output file 1: `YeeLatticeFile/Yee_η.dat`

- The program outputs ε and μ on the η -plane where $\eta = x, y, z$.

- example output file 2: `YeeLatticeFile/Yee_jk.dat`

- The program also outputs $x(l + j/2)$ for $0 \leq l < L$ and $y(m + k/2)$ for $0 \leq m < M$, where $j, k = 0$ or 1 .

- example error file: `logSlice.txt`

- For normal execution, the program outputs data of the input file 1 to this file. It has some message if the program abnormally run.

The input file 2 consists of 2-lines, and an example of the file is shown as follows.

`YeeSlice.txt`:

1-st Data-file name = `YeeLatticeFile/Yee`

The above name is the same as the 2nd line in `YeeCell.txt`.

2-nd `S_u= 12, S_v= 8, S_w= 14`

This file defines slice points $0 \leq S_u < L$, $0 \leq S_v < M$ and $0 \leq S_w < N$ for $x(u)$ -plane, $y(v)$ -plane and $z(w)$ -plane respectively.

[\[Go to table of contents.\]](#) [\[Go to home.\]](#)

5 Simulation group 内の数値処理の流れ

本節では simulation group (図 2.1) に関する入出力ファイルについて例題 (図 3.2) を用いながら説明します。図 5.1 にデータの流れを示します。

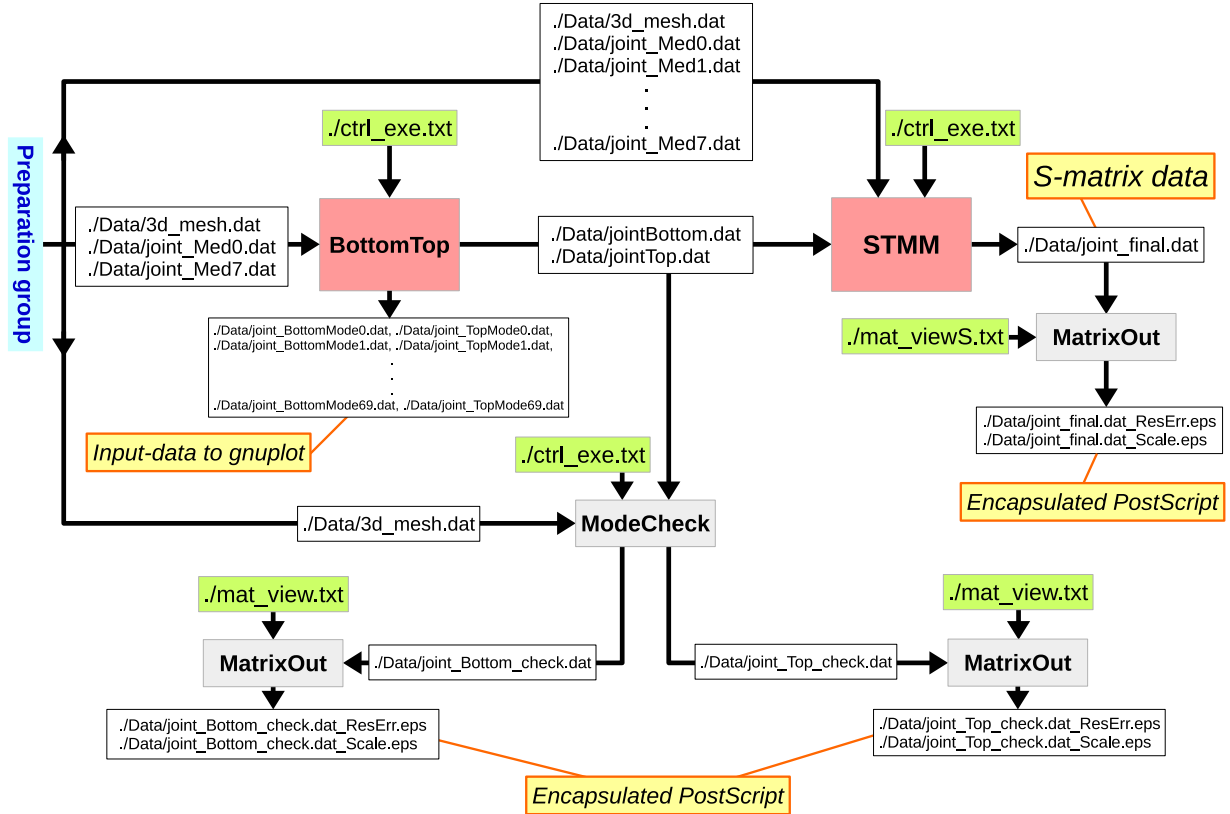


図 5.1: Data flow from BottomTop to STMM in the simulation group. The example in Fig. 3.2 uses MatrixOut in the graphics group at three times. The './Data/joint_final.dat' from STMM shows all elements of S-matrix.

5.1 Mode calculation

This subsection shows a program of generating optical modes for bottom and top waveguides.[?].

- program name: BottomTop.exe
 - source file archive: BottomTop_20120819.zip
 - example batch file: BottomTop.bat
 - example input file 1: Frame_3um4um28p4um.dat
This file is the output file from Mesh3D.exe.

- example input file 2: `YeeCell.txt`

This subsection had been defined for Yee’s lattice generation.

- example output file: `YeeLatticeFile/YeeBottom.dat` and `YeeLatticeFile/YeeTop.dat`

The program outputs $\beta_{\kappa}(j)$ and \mathbf{U}_{κ} when $\kappa = b, t$, and $0 \leq j < LM$.

- example error file: `BottomToplog.txt`

For normal execution, the program outputs data of the input file 1, the input file 2, and execution times to this file. It has some message if the program abnormally run.

The input file 2 is described by commands for optical objects. Maximum line number should be less than 16384 in this file. Definition of each optical object starts at ‘begin’ command, and it finishes at ‘end’ command. Number of the objects should be less than 1024. Lower commands overwrites upper ones in the file. We have to set three commands between ‘begin’ and ‘end’, and we can also use two optional command.

5.2 Stable transfer matrix method

[\[Go to table of contents.\]](#) [\[Go to home.\]](#)

6 Development schedule and history

[\[Go to table of contents.\]](#) [\[Go to home.\]](#)

7 LICENSE AGREEMENT

Copyright (c) 2014, Tatsuya Usuki
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Whenever numerical results obtained from SMatrAn are published in scientific/engineering journals or proceedings, cite the following paper [1]: T. Usuki, M. Saito, M. Takatsu, R.A. Kiehl, and N. Yokoyama, “Numerical analysis of ballistic-electron transport in magnetic fields by using a quantum point contact and a quantum wire,” *Phys. Rev. B* **52**, pp.8244–8255 (1995). If possible, also cite one of the proceedings [2]: T. Usuki, “Frequency domain analysis for optical propagation using Yee lattice,” *Proceedings of 2013 URSI International Symposium on Electromagnetic Theory (EMTS)*, pp.503–505 (2013).

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

参考文献

- [1] T. Usuki *et al.*, Phys. Rev. B **52**, 8244 (1995).
- [2] T. Usuki, Proceedings of 2013 URSI International Symposium on Electromagnetic Theory (EMTS), 503–505 (2013).
- [3] T. Usuki, *unpublished*.
- [4] David Yuk Kei Ko and J. C. T. Inkson, Phys. Rev. B **38**, 9945 (1988).
- [5] Patrick A. Lee and Daniel S. Fisher, Phys. Rev. Lett. **47**, 882 (1981).
- [6] V. T. Renard *et al.*, Phys. Rev. Lett. **100**, 186801 (2008).
- [7] Kjetil M. D. Hals *et al.*, Phys. Rev. Lett. **105**, 207204 (2010).
- [8] Bobo Liu, R. Akis and D. K. Ferry, J. Comput. Electron. **13**, 950 (2014).
- [9] Yoshitaka Watanabe, “Fast”, “Large” and “Accurate” are justice (*in Japanese*).
- [10] For another example, the finite element method (FEM) attains a high-accuracy shape for the solution of a partial differential equation.
- [11] An authority on numerical analysis also commented progress in C99, p.19 of Ichizo Ninomiya *et al.*, ISBN 4-320-12088-4 (2014, *in Japanese*).
- [12] For an example, ‘Mersenne Twister’ was implemented in <random>.