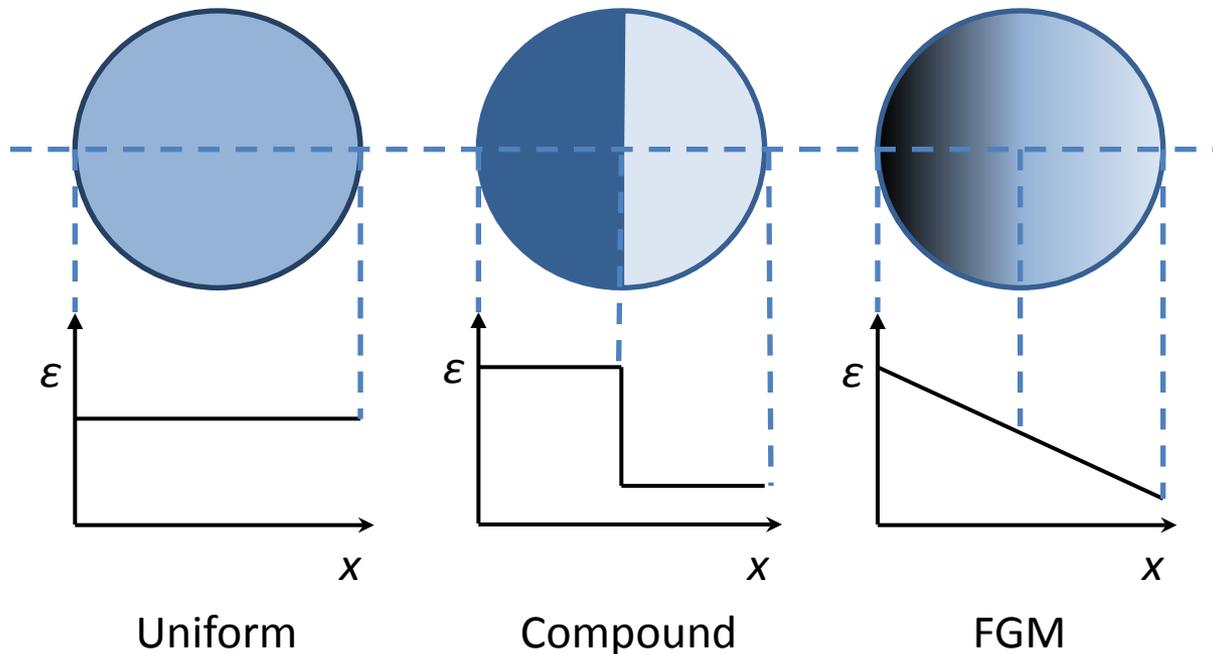


遺伝的アルゴリズムと COMSOL Java API を用いた 分布誘電率推定手法の検討

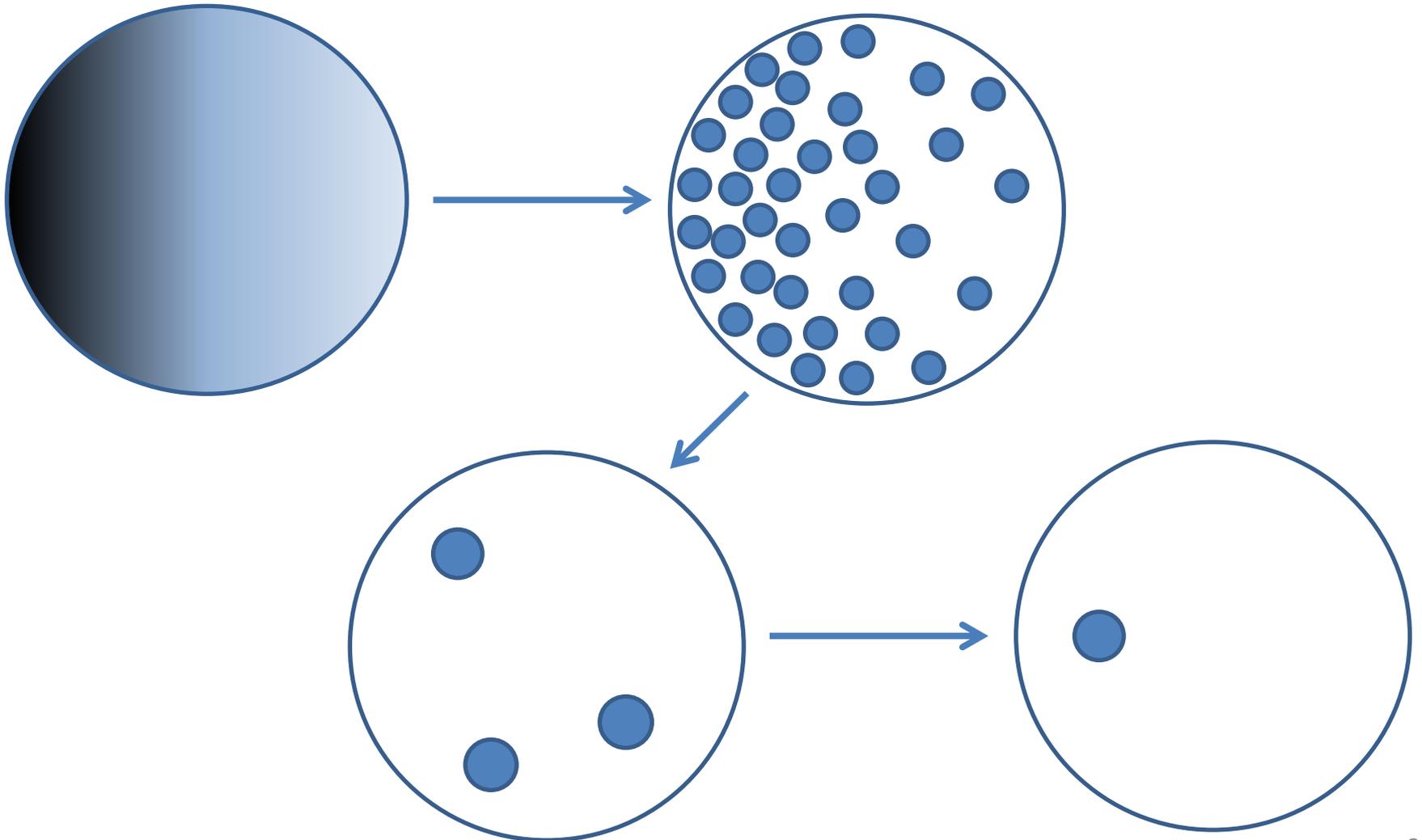
千葉工業大学工学部
電気電子情報工学科
相知研究室 4年
富山 和義

背景

- 傾斜機能材料(FGM)の評価
 - 誘電率が空間的に変化する材料
 - 電力機器の性能向上

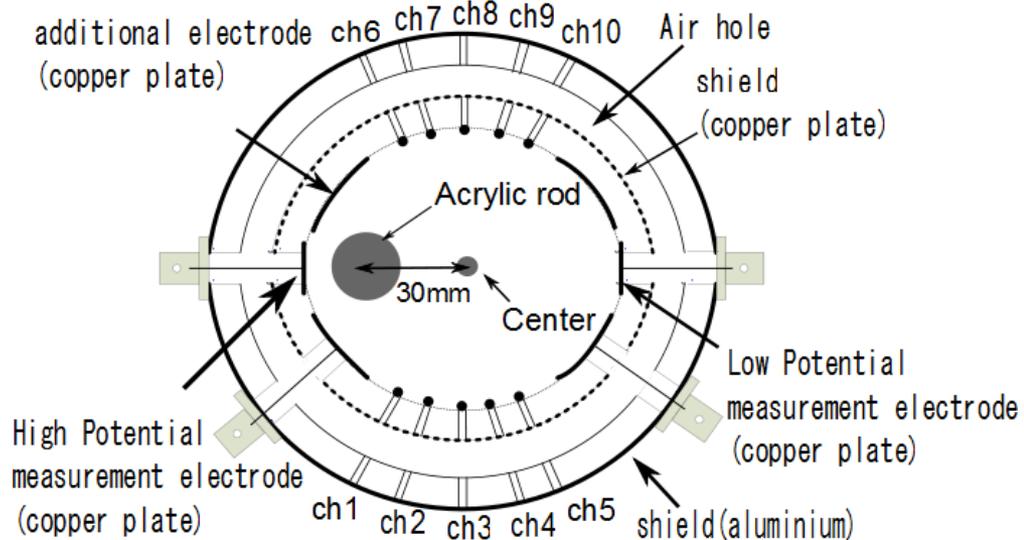
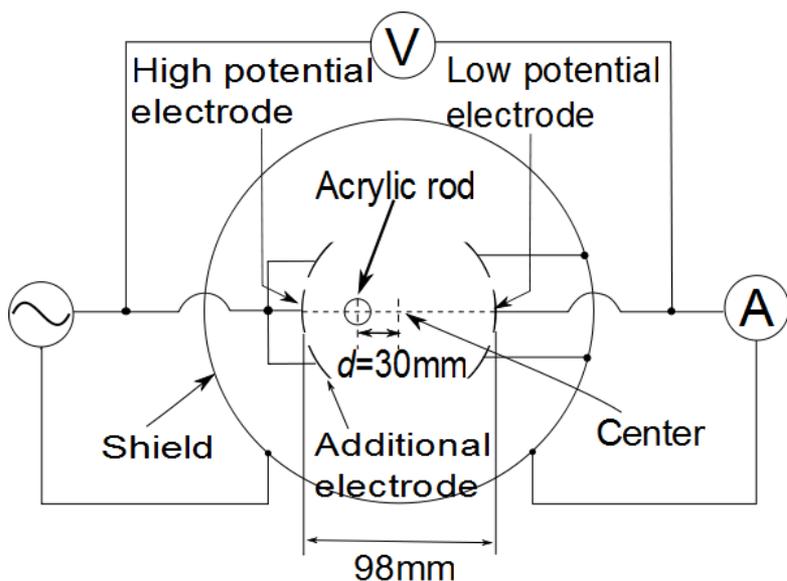


FGM を多数の円柱で模擬する

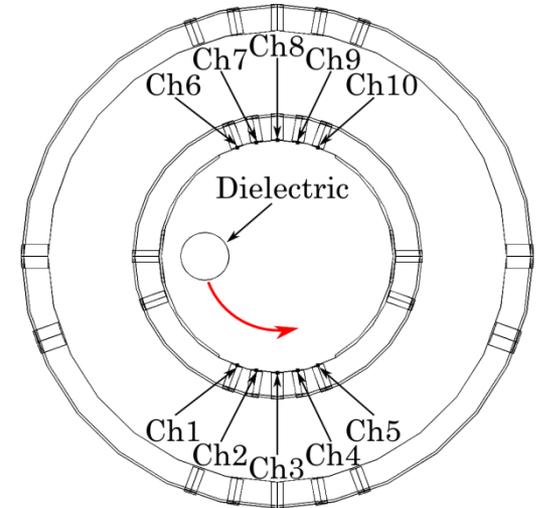
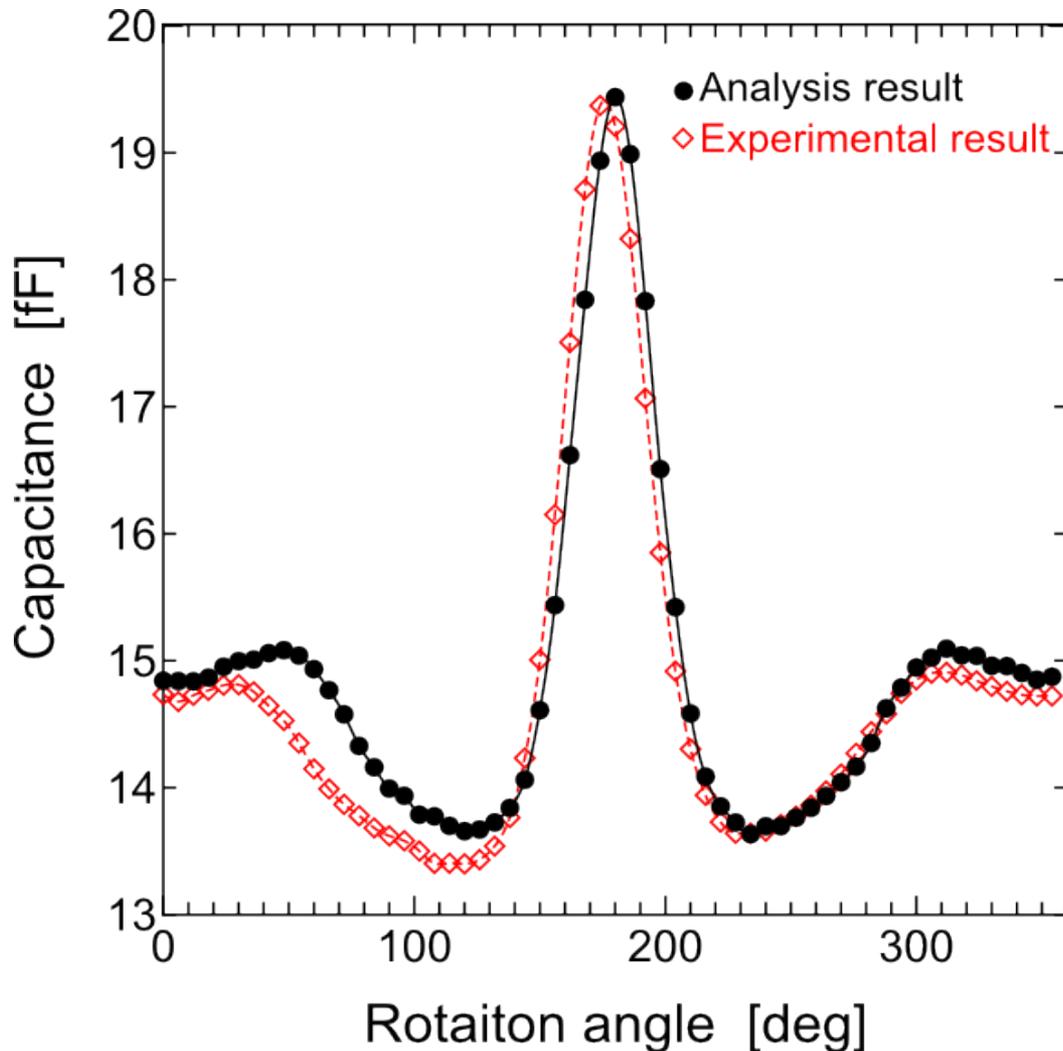


静電容量と電位の変化

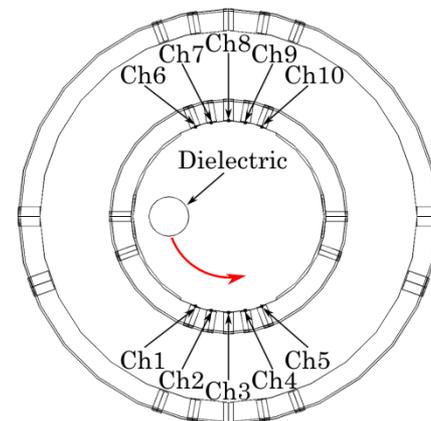
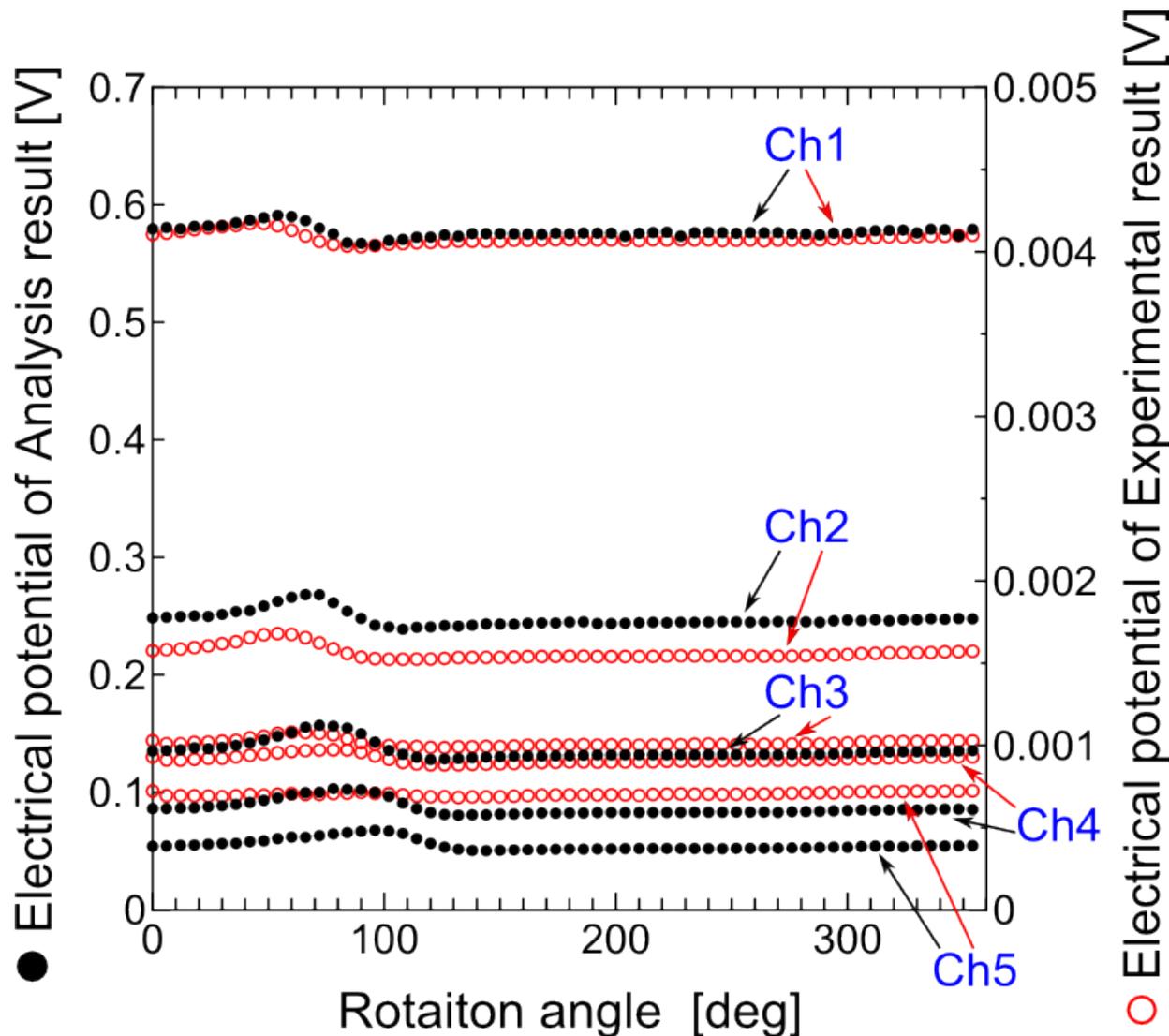
- 電極板の間に被測定物を置き、
回転させると静電容量と電位が変化する



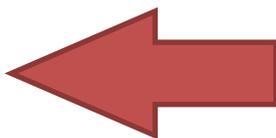
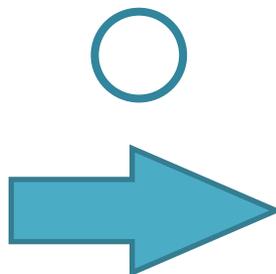
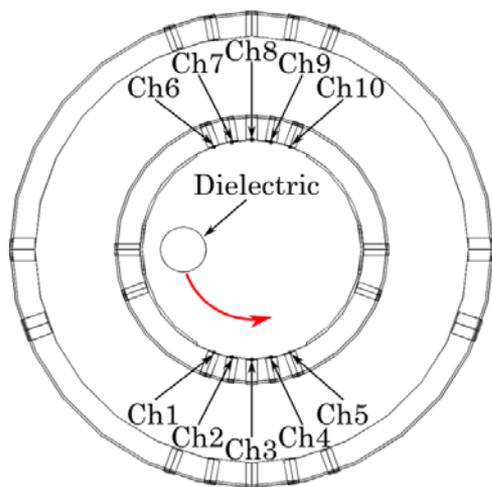
回転させた時の静電容量の変化 (アクリル棒直径：20 mm)



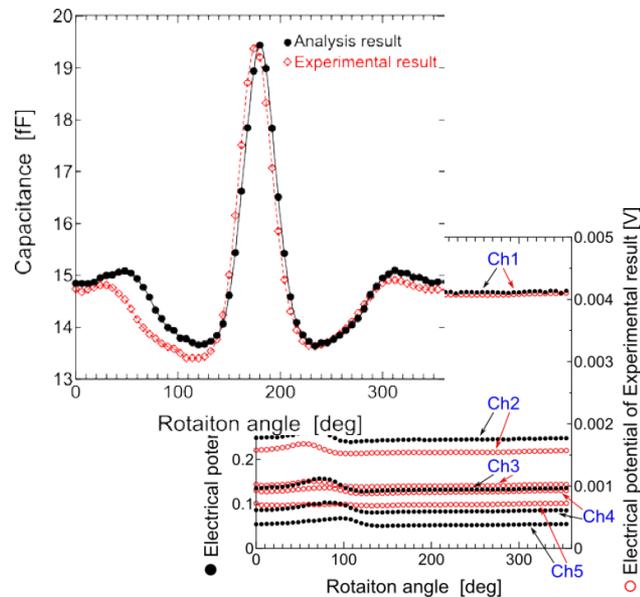
回転させた時の電位の変化 (アクリル棒直径：20 mm)



誘電体の位置，大きさ，誘電率が分かれば，
静電容量や電位は COMSOL で計算できる



?



逆に，静電容量や電位から，
誘電体の位置，大きさ，誘電率を求めたい

遺伝的アルゴリズム

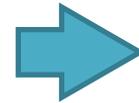
- 生物の進化を模倣したアルゴリズム
- 解の候補を遺伝子の列で表す

誘電体の直径 30 mm

中心からの距離 20 mm

角度 0 deg

比誘電率 3.0



1110000010

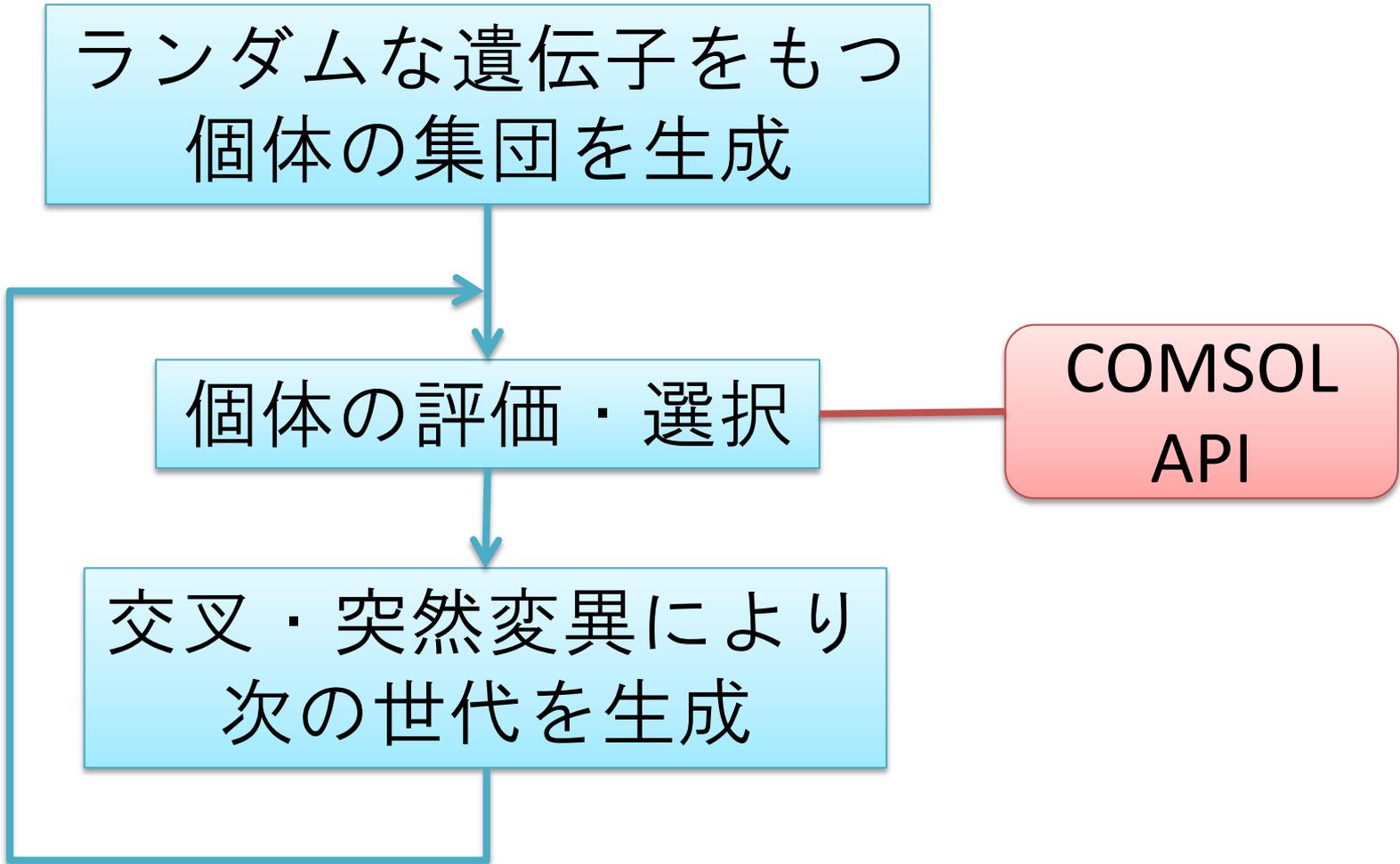
遺伝的アルゴリズム

ランダムな遺伝子をもつ
個体の集団を生成

個体の評価・選択

COMSOL
API

交叉・突然変異により
次の世代を生成



評価と選択

- COMSOL Multiphysics を用いて計算した静電容量や電位の実験値との差
- 適応度の高い固体を多く残し、
適応度の低い固体は淘汰する

A : 適応度 4

B : 適応度 7

C : 適応度 5

D : 適応度 1



B : 適応度 7

B : 適応度 7

C : 適応度 5

A : 適応度 4

交叉・突然変異

- 固体のペアを作り，遺伝子を入れ替える

1101011101 → 1101010010
1110000010 → 1110001101

- 遺伝子をランダムに変化させる

1001011011 → 1011011001

Java 開発環境の導入

- JDK (Java Development Kit)
 - Java 言語のコンパイラ (無料)
 - 公式にサポートされているのは **1.6** まで
 - <http://www.comsol.com/system-requirements>
 - Oracle Java Archive よりダウンロード
 - <http://www.oracle.com/technetwork/java/javasebusiness/downloads/java-archive-downloads-javase6-419409.html#jdk-6u45-oth-JPR>

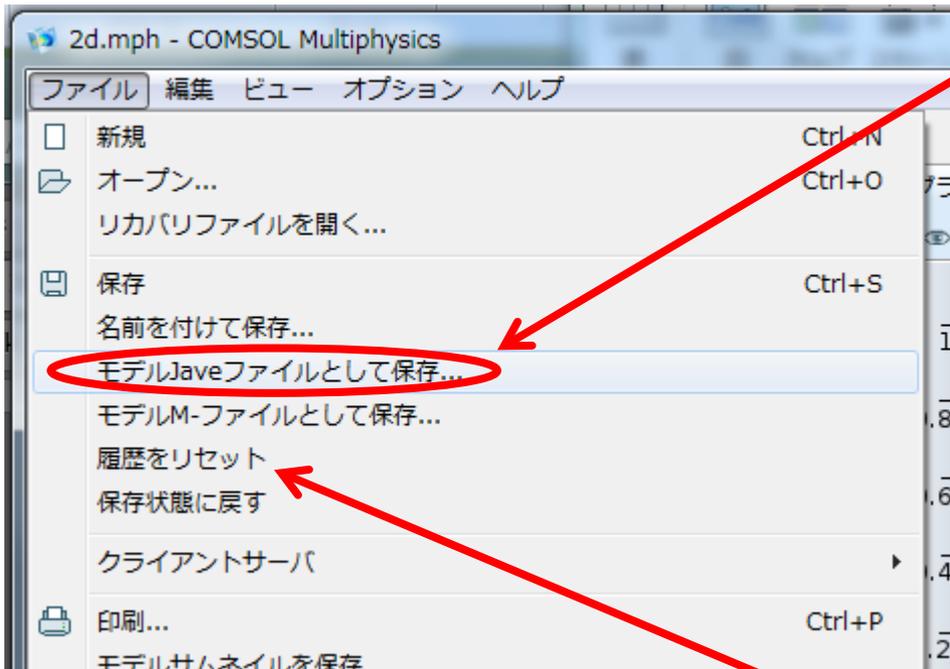
Java 開発環境の導入

- NetBeans IDE
 - 統合開発環境
 - COMSOL API を使用するためのライブラリの指定などが容易なため
 - <https://netbeans.org/downloads/7.3.1/>

プログラムの作成

- COMSOL API を調べる
 - GUI で作成したモデルを Java で保存
 - API Reference Manual を読む
 - インストールフォルダ内にあります
 - C:\Program Files\COMSOL\COMSOL43b\doc\pdf\COMSOL_Multiphysics\COMSOL_APIReferenceManual.pdf

モデルを Java ファイルに保存



「モデル Java ファイルとして保存」すると，GUIで操作した順番で Java のソースコードが出力されます

先に「履歴をリセット」をするとソースコードが整理されます

出力されるコード例

```
public class AcrylColumn {  
    public static void main(String[] args) {  
        run();  
    }  
    public static Model run() {  
        Model model = ModelUtil.create("Model");  
        :  
        :   ここに GUI で操作した内容に対応  
        :   したコードが出力されます  
    }  
}
```

出力されるコード例

- 回転コピー 3 (mir3)
- 外部の穴2 (r10)
- 回転 10 (rot10)
- 鏡面コピー 4 (mir4)
- 差 7 (dif7)
- 空間 (sq1)
- アクリル棒 (c13)**
- 回転 11 (rot11)
- 電位測定電極 (pt1)
- 回転 12 (rot12)
- 回転 13 (rot13)
- 回転 14 (rot14)
- 回転 15 (rot15)
- 鏡面コピー 5 (mir5)

▼ サイズおよび形状

半径: m

セクタ角: deg

▼ 位置

ベース:

x: m

y: m

▼ 回転角

回転: deg

出力されるコード例

ジオメトリ 1

円 13

```
model.geom("geom1").feature("c13").set("r", "0.01");
```

半径を

0.01 m に設定

```
model.geom("geom1").feature("c13")  
.set("pos", new String[]{"-0.03", "0"});
```

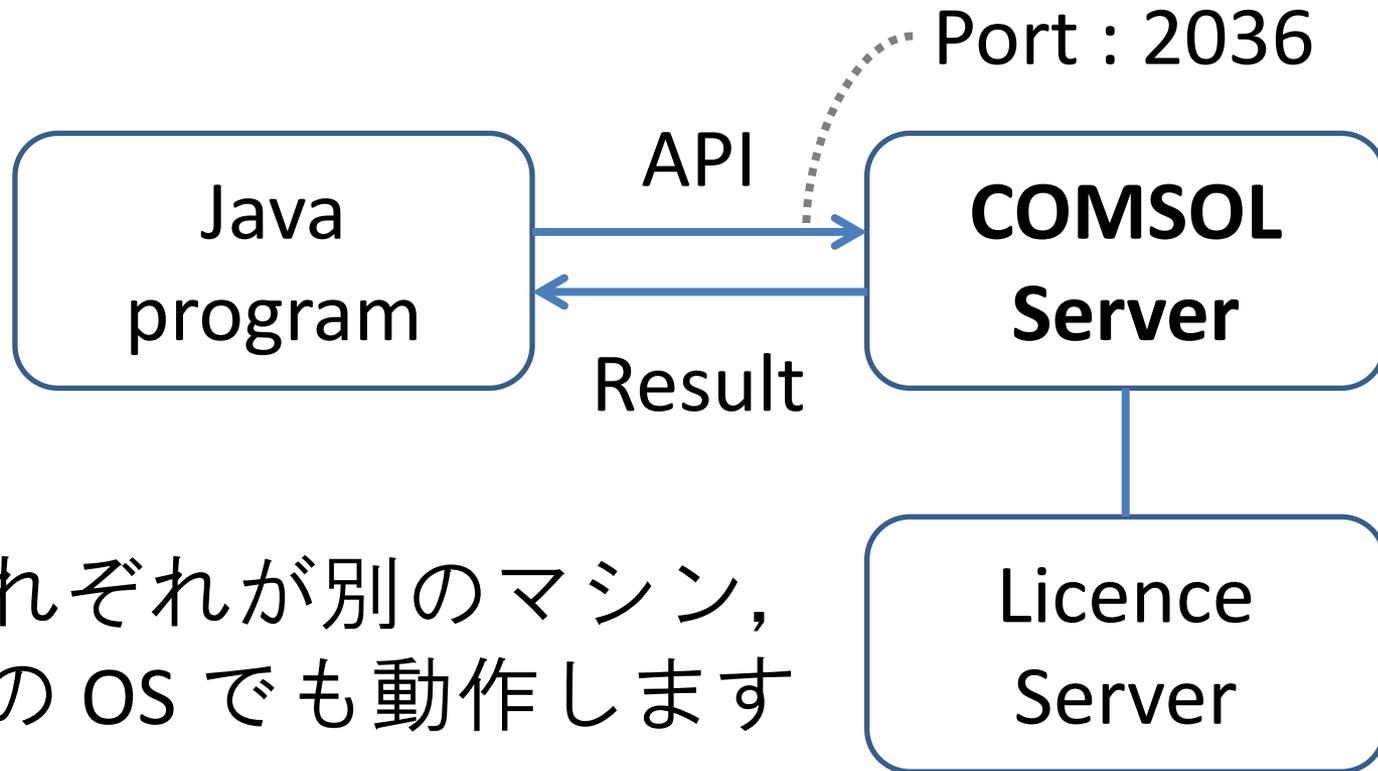
位置を

x = -0.03 m

y = 0 m

に設定

COMSOL Server



それぞれが別のマシン,
別の OS でも動作します

接続と切断

```
try {  
    // 接続  
    ModelUtil.connect("localhost", 2036);  
    run();  
} catch (Exception e) {  
    // エラー処理  
}  
finally {  
    // 切断  
    ModelUtil.disconnect();  
}
```

ホスト名 ポート番号



結果を取得する

// テーブル 2 の内容を取得する

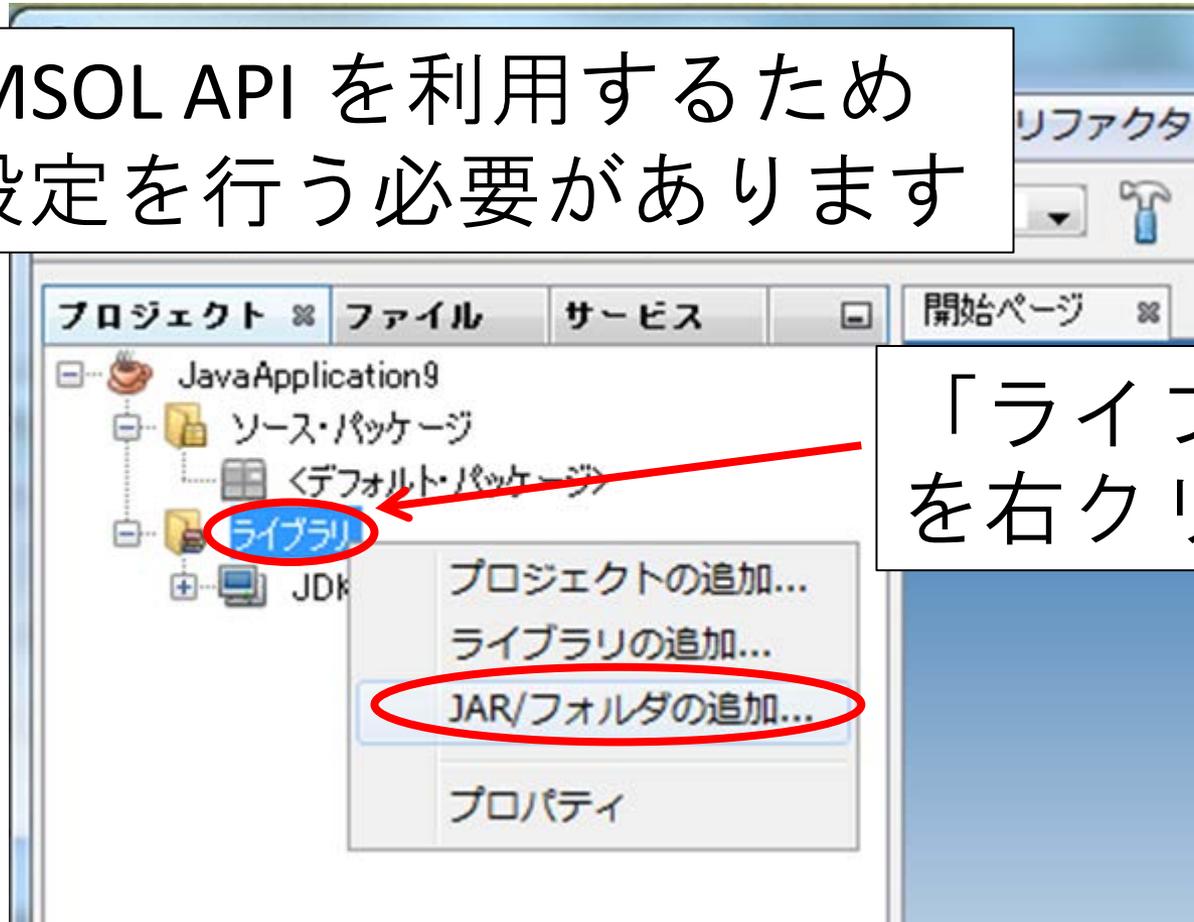
```
String[][] tbl2 = table("tbl2").getTableData(true);
```

電位 (V), ポイント: 75	電位 (V), ポイント: 76	電位
0.6203164688607171	0.6201355042419732	0.29

true:フル精度

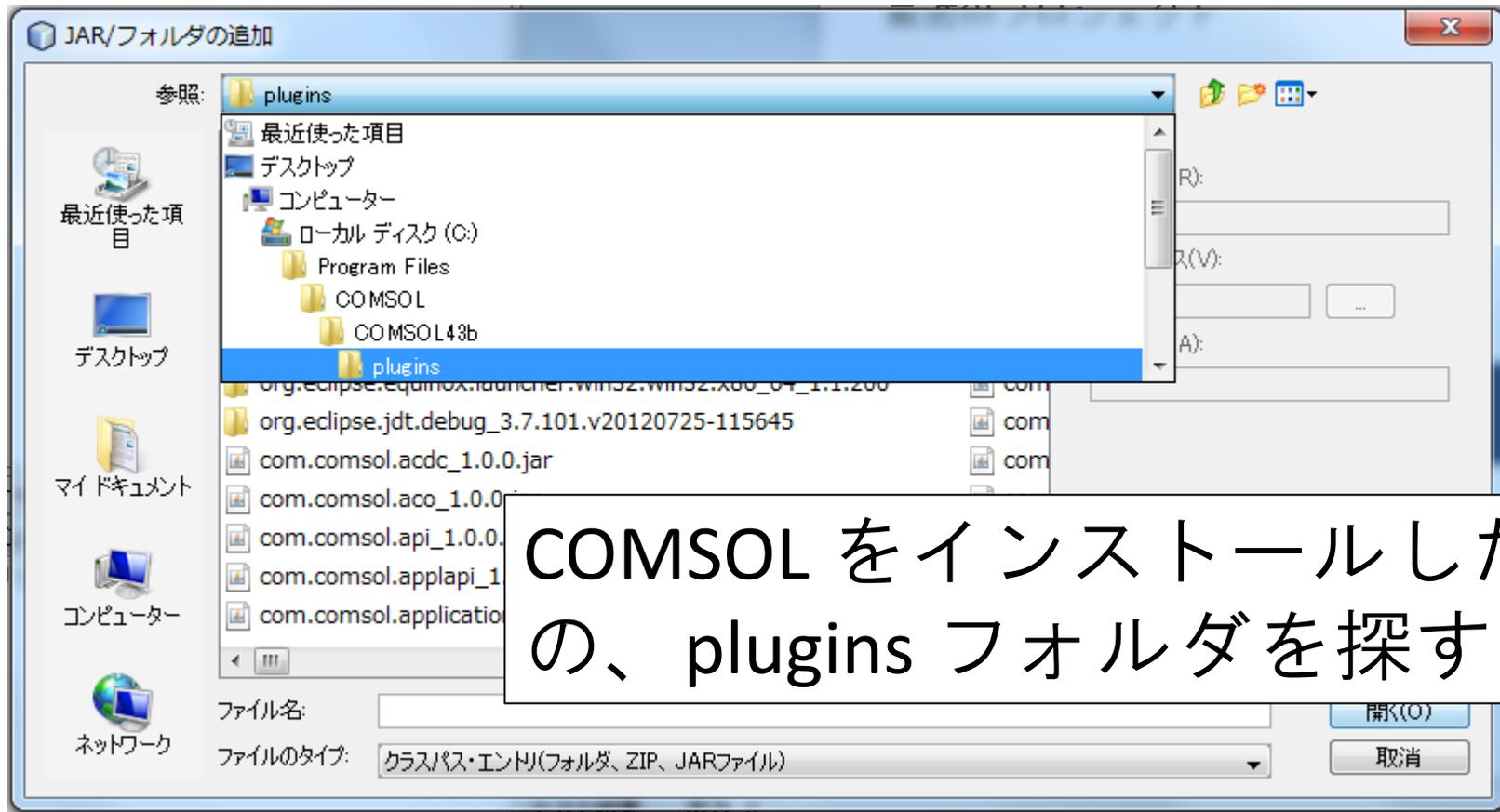
ライブラリ (JAR) の指定

COMSOL API を利用するために設定を行う必要があります



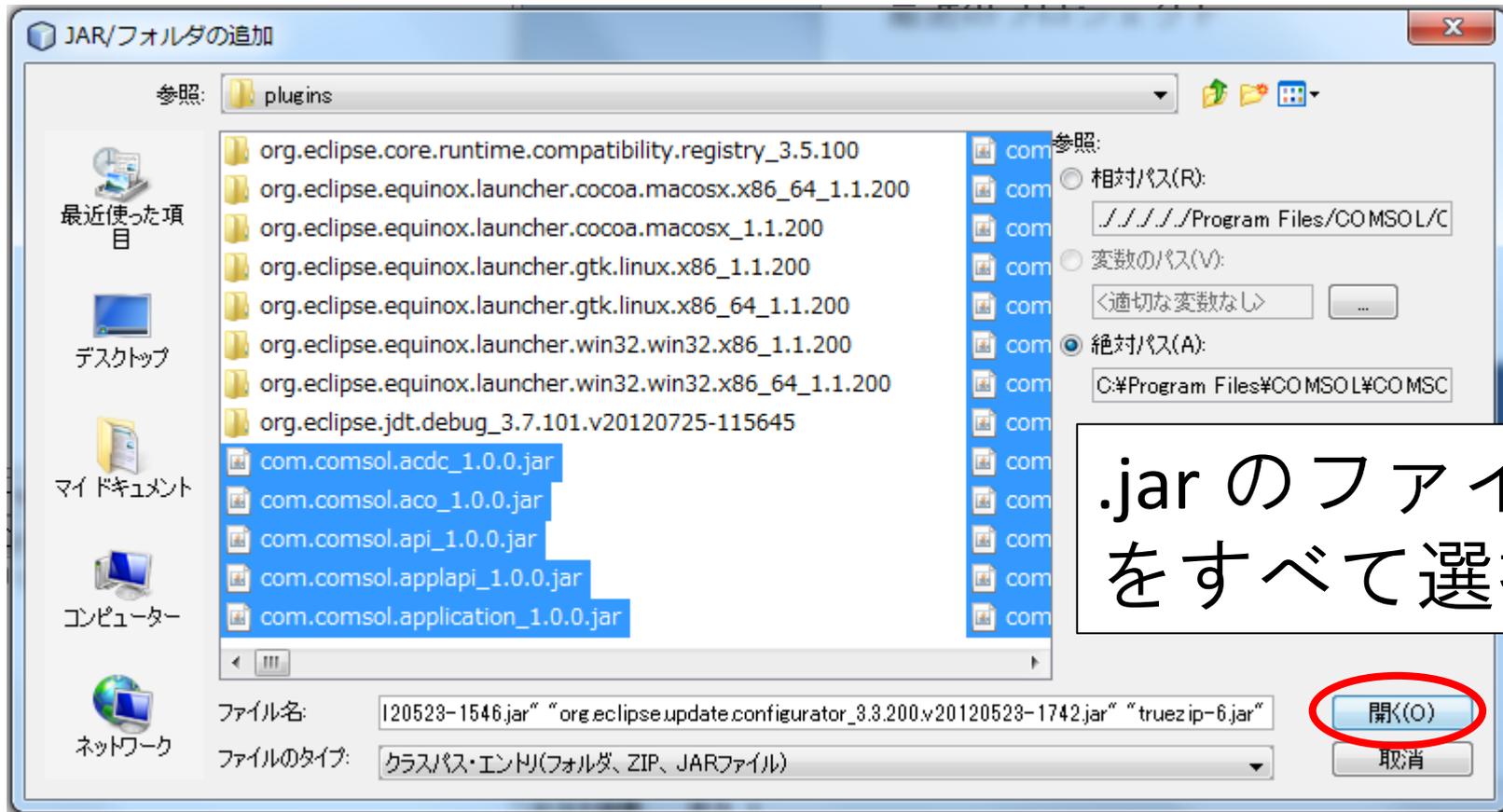
「ライブラリ」
を右クリック

ライブラリ (JAR) の指定

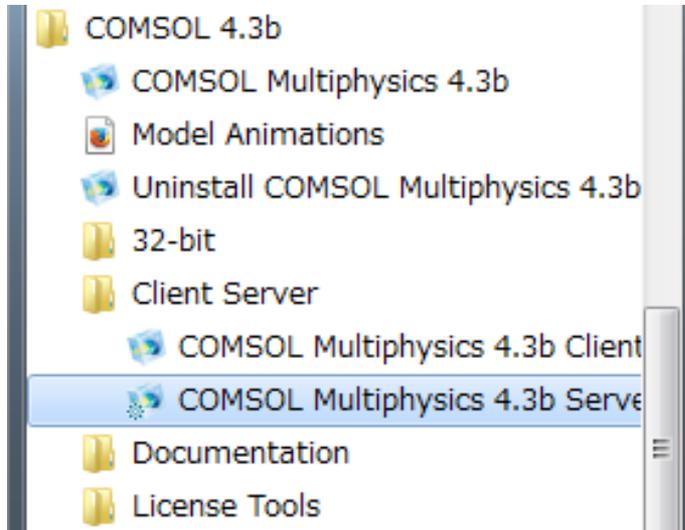


COMSOL をインストールした場所の、plugins フォルダを探す

ライブラリ (JAR) の指定



COMSOL Server の実行



Windows の場合

初めて起動するとユーザ名とパスワードの設定を求められるので入力します

Linux の場合

/(インストール先)/bin/comsol server
(/usr/local/comsol43b/ など)

プログラムの実行

NetBeans で「プロジェクトの実行」をします

