CommonMP 計算実行環境 KyotoUnivEngHywrCMPEE 解説書 Version 1.3

立川康人・高橋 円

平成 25 年 1 月 20 日

概 要

KyotoUnivEngHywrCMPEE、略して CMPEE, CommonMP Execution Environment は、CommonMP の構造定義ファイル (XML 形式) に記述されている要素モデルのパラメータを変更し、CommonMP のコマンドライン実行環境 hymco.exe を用いて計算を実行するフロントエンドである。1) 任意の要素モデルのパラメータの一覧表示機能、2) 一覧表示したパラメータを変更して計算する機能、3) 一連の計算をある書式に従って記述し、それに従って連続的に計算する機能、4) 最適モデルパラメータを求める機能、を備えている。

目 次

1 はじめに **2**

2 使用前の準備 **3**

3 起動画面 **3**

3.1 メニューバー 4
3.2 要素モデル名表示欄 5
3.3 プロジェクトファイル名表示欄 5
3.4 パラメタ名・値表示画面 5
3.5 計算方法の選択ボタン 5

4 パラメータ値の一覧表示と変更および計算手順 **7**

5 計算手順指定ファイルを用いたバッチ処理の手順 **8**

5.1 計算の手順 8
5.2 計算手順指定ファイルの仕様 8

6 パラメータ同定計算の手順 **11**

1 はじめに

個々の水理・水文計算を実現する数値計算プログラムは数多く存在する。それらの数値計算プログラムを相互に組み合わせて利用することができれば、高度な水理・水文計算を実施することができる。しかし，個々の数値計算プログラムは通常，異なる組織に属する技術者・研究者によって独立に開発されるため、計算機環境やプログラミング言語、計算機プログラムの構成の仕方が異なり、それらを相互に組み合わせて利用することは容易ではない。このことは、複雑で高度なシミュレーションモデルになればなるほど、それを開発した個人か研究グループしかそのソースコードを理解することができず、モデル構造や計算アルゴリズムを評価することが難しいこと、他者が開発した要素モデルを組み合わせて、高度で複雑な水理・水文シミュレーションを構築することが困難であることを意味する。こうした課題を解決するために、国土交通省国土技術政策総合研究所が中心となって、水理・水文解析ソフトウェア統合型共通基盤 CommonMP(Common Modeling Platform for water-material circulation analysys)[1] が開発された。

要素モデルを組み合わせて高度なシミュレーションモデルを短時間で構築する環境が、CommonMP により提供された。水工シミュレーションモデルを実際の現場に利用していくためには、モデル構築環境と合わせて、CommonMP に備わっていない以下の機能、

- 多数の要素モデルのパラメータ値や初期状態量の値を容易に変更する機能
- パラメータ値や初期状態量の値、入力ファイルを変えて、多数の水工シミュレーションを連続的に実行する機能
- モデルパラメータを最適同定する機能

を持つ実行環境が必要になる。一つ目の機能は、CommonMP の GUI 環境で要素モデルごとにプロパティ画面を開き、パラメータ値を設定する作業を回避するために必要な機能である。要素モデルの個数が多くなると、パラメータ値や初期状態量の変更が膨大な作業となってしまう。二つ目の機能は、パラメータ値や初期状態量の値を変えて計算結果の感度分析を行う場合など、様々な条件での水工シミュレーションを連続的に実行するための機能である。三つ目の機能は、CommonMP を用いてパラメータの最適同定を行う機能である。

CommonMP 計算実行環境 (KyotoUnivEngHywrCMPEE、略して CMPEE, CommonMP Execution Environment) は、これらの 3 つの機能を CommonMP のコマンドライン実行環境を用いて実現するフロントエンドプログラムであり、Visual C# を用いて開発されている。CMPEE は、CommonMP の構造定義ファイル (XML 形式) に記述されている要素モデルのパラメータ値や初期状態量の値を自在に変更することにより、

1) 任意の複数の要素モデルのパラメータを表形式で一覧表示する機能、
2) 一覧表示した要素モデルのパラメータ値を一括で変更して計算する機能、
3) 複数の異なる計算条件をある書式に従ってファイルに記述し、その手順に従って複数のシミュレーションを連続的に実行する機能、
4) 最適モデルパラメータを求める機能

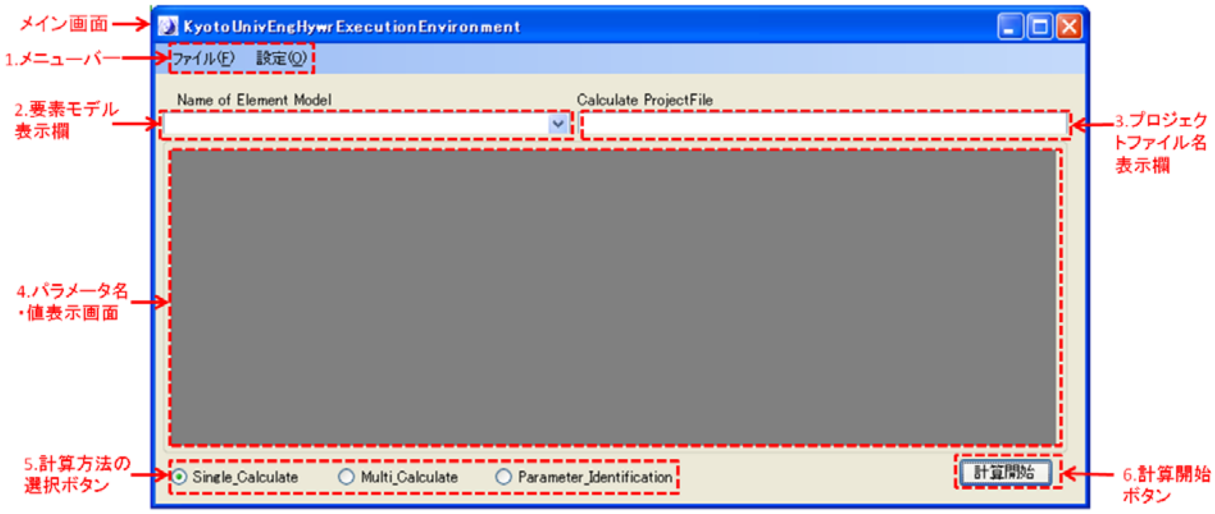


図 1: 起動画面

を実現している。最適モデルパラメータを求めるための最適化アルゴリズムとして、SCE-UA (Shuffled Complex Evolution method developed at the University of Arizona) を用いている。

2　使用前の準備

本プログラムを起動する前に以下を行う。

1. 本プログラム CMPEE.exe を CommonMPMain.exe と同じフォルダに置く。
2. SCE-UA.exe を CommonMPMain.exe と同じフォルダに置く。SCE-UA.exe はパラメータ同定計算に用いる。SCE-UA.exe を cygwin でコンパイルして作成した場合は、cygwin1.dll も同じフォルダにコピーする。cygwin1.dll は cygwin のランタイムライブラリであり、cygwin の /bin の中にある。
3. CommonMP/Execute/conf/HymcoCUI.cfg をテキストエディタで修正して、プロジェクトファイルおよび構造定義ファイルをフォルダ "CUI 作業ワークディレクトリ" をフルパスで指定する。CommonMP 1.20 ではフォルダの変更設定が効かないようだ。この場合はデフォルトの Execute/temp に置く。

3　起動画面

起動画面は CMPEE.exe の起動時に表示される画面である。**図 1** に起動画面を示す。以下、起動画面について説明する。

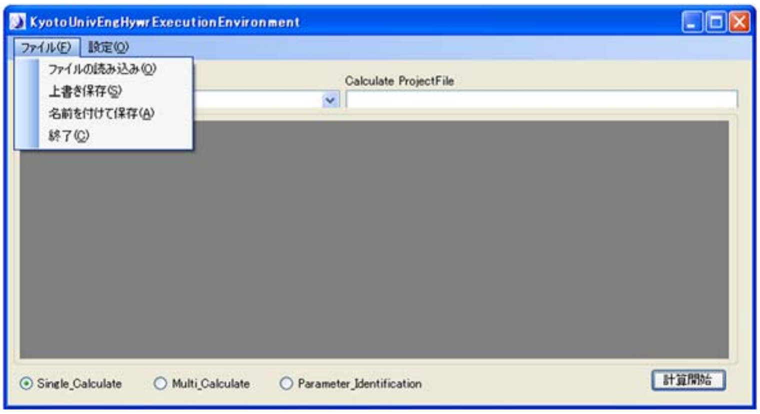


図 2: ファイルメニューのイメージ

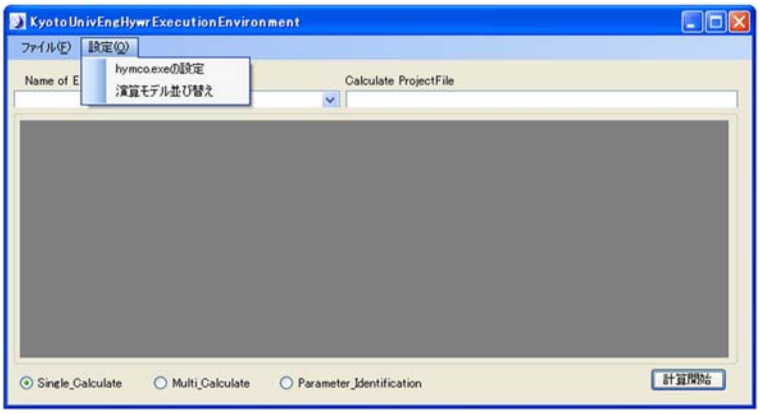


図 3: 設定メニューのイメージ

3.1 メニューバー

ファイルメニューと設定メニューのイメージと項目を**図 2**、**図 3**、**表 1**、**表 2** に示す。起動画面の「設定メニュー」-「要素モデルの並び替え」についての説明を行う。構造定義ファイルを CommonMP の GUI から作る場合、要素モデルは画面上に設置した順番にファイルに表記される。本プログラムが要素モデルのパラメタを表記する際は、構造定義ファイルに表記されている順番を元に表示するため**図 4**(a) のように使用者が求める順番になっていない可能性がある。そこで、並び替える機能を用いることで使用者が求める順番に構造定義ファイルを書き換え、**図 4**(b) のように表示することが出来る。

本機能は**図 4**(a) のように CommonMP の GUI で設定された要素モデルの名称を用いる。要素モデルの名称は構造定義ファイルでも設定することができる。要素モデルに与えた名称が "Hoge(XX)" の形式で書かれている場合、**図 4**(b) のように構造定義ファイルを書き換える。ただし Hoge にはモデル名を、XX には整数が入り、整数の順番に並べ替える。さらに本プログラムの仕様により、名称の最初にある () 内の数字を読み込む。例えば"Hoge(Hoge)(1)"は並び替えの対象とみなさない。また要素モデルを並び替え、それを構造定義ファイルに反映する際は "NewSCF.xml" というファイル名で構築される。

表 1: ファイルメニューの内容

項目名	内容
ファイルの読み込み (O)	CommonMP の構造定義ファイルを読み込む
上書き保存 (S)	構造定義ファイルを更新する
名前をつけて保存 (A)	新しい構造定義ファイルを作成し、保存する
終了 (C)	プログラムを終了する

表 2: 設定メニューの内容

項目名	内容
hymco.exe の設定	CommonMP を CUI で起動させる hymco.exe のディレクトリの設定を行う
要素モデルの並び替え	構造定義ファイル内の要素モデルを指定した順番に並び換える

3.2 要素モデル名表示欄

要素モデル名表示欄には構造定義ファイルに表記されている全ての要素モデル名が格納される。パラメタ値を表示したい要素モデル名を選択することでパラメタ値が表示される。

3.3 プロジェクトファイル名表示欄

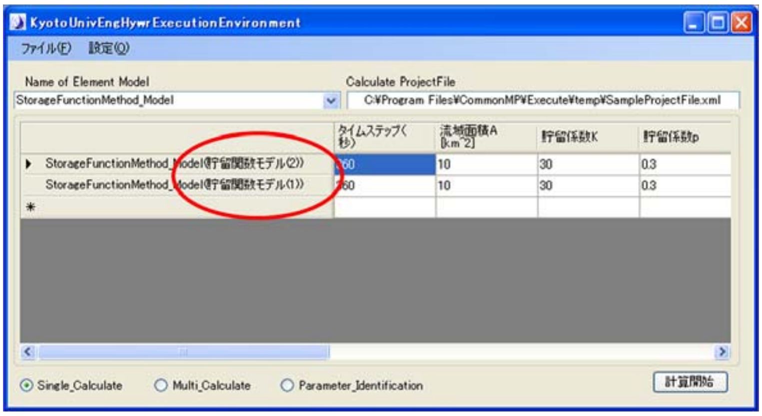
プロジェクトファイル名表示欄には計算を行うプロジェクトファイル名が表示される。デフォルトでは読み込んだ構造定義ファイルのファイル名の語尾を”.xml”から”ProjectFile.xml”に換えたものが表示される (例えば、Sample.xml を読み込んだ場合 SampleProjectFile.xml が表示される)。また新しい名前を付けて保存を行った場合は、新しいファイル名の語尾に対して同様のことを行い表示する。計算を行う際にはこのプロジェクトファイル名表示欄に表記されているプロジェクトファイルを用いる。

3.4 パラメタ名・値表示画面

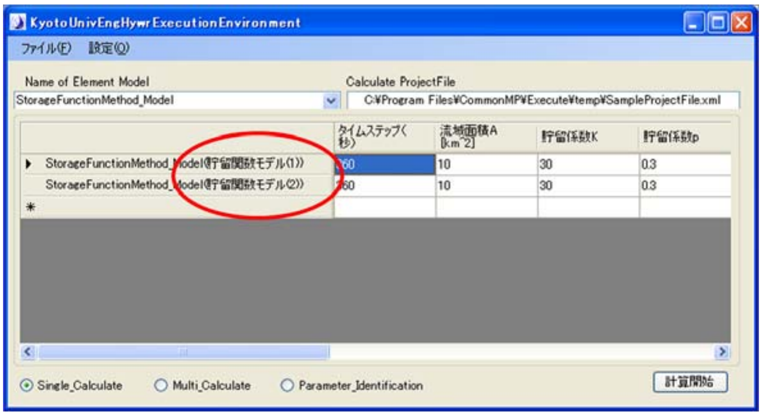
要素モデル名表示欄に表示された要素モデル名に設定されているパラメタ名・値を表示する画面である。なお、VisualStudio2008 によって開発したアプリケーションには貼り付け機能が備わっていないため、下記ソースコードにより、キーボードの ”Ctrl+V” で貼り付けを行う機能を追加した。

3.5 計算方法の選択ボタン

本プログラムには三つの機能が備わっており、これらのラジオボタンより機能を選択することができる。以下、それぞれのラジオボタンについて説明する。



(a) 並び替え前



(b) 並び替え後

図 4: 構造定義ファイルの並び替え

”Single_Calculate”ラジオボタン ”Single_Calculate”ラジオボタンにチェックがある場合は、プロジェクトファイル名表示欄に表示されているプロジェクトファイルを用いて計算を一回行う。操作手順については **4** で説明を行う。

”Multi_Calculate”ラジオボタン ”Multi_Calculate”ラジオボタンにチェックがある場合はパラメタの値を変えながら連続で計算を行う。連続計算を行う手順は **5** で説明する。

”Parameter_Identification”ラジオボタン ”Parameter_Identification” ラジオボタンにチェックがある場合はパラメタの同定を自動で行う。チェックを入れた後は同定を行うパラメタの選択を行い、その初期値などを設定し計算を開始する。パラメタ同定を行う手順は **??** で説明を行う。

”計算開始”ボタン 計算を開始するボタンである。”Multi_Calculate”ラジオボタンのチェックがある場合は連続計算を行い、”Parameter_Identification”ラジオボタンのチェックがある場合はパラメタ同定の計算を行い、いずれにもチェックが無い場合は、一

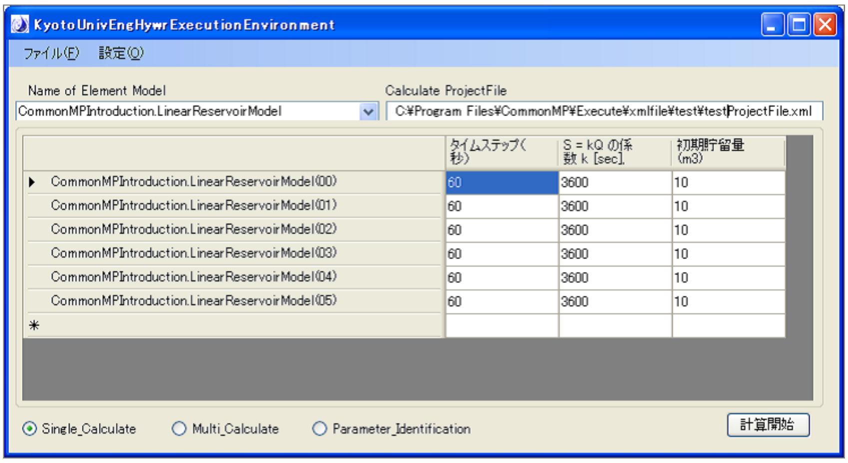


図 5: パラメタの一覧表示

回の計算が行われる。

4 パラメータ値の一覧表示と変更および計算手順

「ファイル」-「ファイルの読み込み」から構造定義ファイル (XML 形式) を選択することで、**図 5** のようにパラメタ名・値表示画面にパラメタ名と値が表示される。**プロジェクトファイルではなく、構造定義ファイルを読み込むことに注意する。**また、構造定義ファイルとプロジェクトファイルは CommonMP/Execute/conf/HymcoCUI.cfg で設定した "CUI 作業ワークディレクトリ" に置くこと。ここに置かないとコマンドライン環境での CommonMP である hymco.exe が正常に動作しない。デフォルトフォルダは Execute/temp である。

図 6 のように要素パラメータの値を変更することができる。演算要素モデルを選択すれば、モデルパラメータの値や初期状態量を変更することができる。出力用要素モデルや入力用要素モデルを選択すれば、出力ファイルや入力ファイルのフォルダやファイル名を変更することができる。

パラメタの変更を行った後は、「ファイル」-「上書き保存」、もしくは「ファイル」-「名前を付けて保存」からパラメタを修正した構造定義ファイルを保存することができる。パラメータを修正して新たに計算する場合は、構造定義ファイルを保存した後、実施すること。保存しないと修正したパラメータ値が計算に反映されない。計算に用いられるプロジェクトファイルは、プロジェクトファイル名表示欄に表示されているプロジェクトファイルである。

計算に用いられる CommonMP の CUI である hymco.exe は「登録」-「hymco.exe の登録」から設定することができる。デフォルトでは "C:/Program Files/CommonMP/Execute/bin/hymco.exe" を設定している。パラメータの更新などが完了した後に、"計算開始" ボタンを押せば、hymco.exe による計算が行われる。

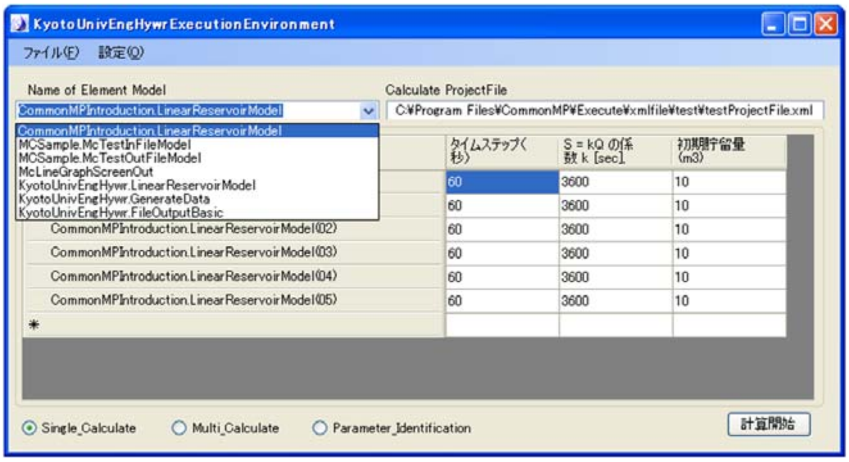


図 6: 要素モデルの選択

5　計算手順指定ファイルを用いたバッチ処理の手順

5.1　計算の手順

条件の異なる複数の計算を連続して実施するバッチ処理について説明する。変更するパラメータ値や入力ファイル、出力ファイル名を記述する、所定の形式に従うテキストファイルを用意し、この計算手順に従って一連の計算を一度に実施する。この形式の計算手順を指定するファイルを「計算手順指定ファイル」とよぶことにする。

計算手順指定ファイルを用意した後、**図 7** のように ”Multi_Calculate” ラジオボタンにチェックを入れた後に”計算開始” ボタンを押すと、上記ファイルの読み込みを行い、一連の計算が実施される。

計算手順指定ファイルでは、**プロジェクトファイルのフルパス**を指定する。構造定義ファイルの読み込みは必要ない。仮に読み込んだとしても、バッチ処理による計算には影響しない。プロジェクトファイルを計算ごとに指定し、そのプロジェクトファイルをもとにパラメータ値などの変更を設定して計算をする仕様となっている。

連続した計算ではログを出力する。ログは hymco.exe と同じフォルダに出力される。ファイル名は ”multi_log.txt” である。出力要素モデルによるファイル出力先のフォルダは、自動的に作成される。ただし、環境によってはフォルダの自動生成が有効でないことがあるため、その場合は利用者があらかじめフォルダを用意する必要がある。

5.2　計算手順指定ファイルの仕様

計算手順指定ファイルの拡張子は ”.data” または ”.spf” とする。計算手順指定ファイルは [ ] で書かれた固有の説明行の後に情報を記述する。一回分の計算の情報は [Calculate_start] と [Calculate_end] の間に記述する。計算手順指定ファイルはどのフォルダにあっても構わない。行頭に # がある行はコメント行とみなされる。改行のみからなる空行は読み飛ばされる。計算手順の情報は以下の順番に記述する。

[Calculate_start]

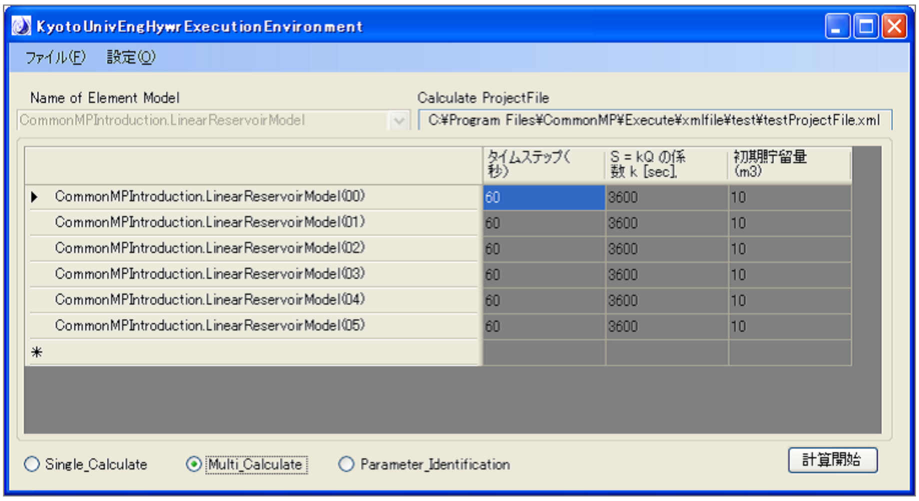


図 7: ”Multi_Calculate”ラジオボタンをチェック時の画面

一回分の計算手順の開始を表す。

[Xml_Directry]

対象とするプロジェクトファイル（構造定義ファイルではない）のフルパスを記述する。

[Model_name_of_parameter] パラメータ値の変更したい要素モデルの名称を記述する。具体的には以下の構造定義ファイルの中の

Model=”KyotoUnivEngHywr.StorageFunctionMethod”

がモデル名称である。

ソースコード 1: 要素モデルのモデル名称

```
<Components>
  <CalElement ID="E38-15115-792012"
    Model="KyotoUnivEngHywr.StorageFunctionMethod"
    FactoryID="KyotoUnivEngHywr.StorageFunctionMethod.Factory"
    ShapeKind="Basin" X="-35" Y="-69" Rotation="0" Width="20" Height="20"
    ElementName="">
      ...
  </CalElement>
</Components>
```

[Parameter_value] 演算要素モデルのパラメタ値や入・出力要素モデルのファイル名を記述する。変更しないパラメータも含めて、すべてのパラメータを記述する必要がある。具体的には、構造定義ファイルの中の上で指定する要素モデルの中に記述されている

- <Property> 情報 (パラメータ情報) で指定する <Variable Key= ... Value=”300” /> などとなっている Value で指定されている値
- <InitialSet> 情報 (初期状態情報) の <Variable Key= ... Value=”2” /> などとなっている Value で指定されている値

を並べて指定する。構造定義ファイルの Variable Key に対応するすべての値を、記載順に、空白あるいはタブで区切って指定する。変更しない場合も値を指定しなくてはならない。

ソースコード 2: 要素モデルのパラメータ値および初期状態値

```
<Property>
  <Variable Key="TIME_STEP" Name="タイムステップ（秒）" DataType="DOUBLE" Value="300" />
  <Variable Key="m_dArea" Name="流域面積␣A␣[km^2]" DataType="DOUBLE" Value="100" />
  <Variable Key="m_dK" Name="貯留係数␣K" DataType="DOUBLE" Value="30" />
  <Variable Key="m_dP" Name="貯留係数␣p" DataType="DOUBLE" Value="0.6" />
  <Variable Key="m_dTl" Name="遅滞時間␣TL␣[hr]" DataType="DOUBLE" Value="0" />
  <Variable Key="m_dF1" Name="1␣次流出率␣F1" DataType="DOUBLE" Value="0.7" />
  <Variable Key="m_dRsa" Name="飽和雨量␣Rsa␣[mm]" DataType="DOUBLE" Value="50" />
  <Variable Key="m_dR0" Name="初期損失雨量␣R0␣[mm]" DataType="DOUBLE" Value="0" />
  <Variable Key="m_dQb" Name="基底流量␣Qb␣[m^3/s]" DataType="DOUBLE" Value="10" />
</Property>
<InitialSet>
  <Variable Key="m_dQ1" Name="初期流出高 (mm/hr)" DataType="DOUBLE" Value="2" />
</InitialSet>
}
```

[Output_Directry_Change] 出力要素モデルの出力先を変更する。要素モデル名、出力先が書かれている Variable の Key 名、新しい出力先ディレクトリを 3 行に分けて表記する。ここではディレクトリのみを変更するため、ファイル名の箇所には「*」を入れる。

[Input_Directry_Change] 入力要素モデルの読み取り先を変更する。要素モデル名、読み取り先が書かれている Variable の Key 名、新しい読み取り先ディレクトリを 3 行に分けて表記する。ここではディレクトリのみを変更するため、ファイル名の箇所には「*」を入れる。

複数の要素モデルのパラメタ値を変更する場合は [Model_name_of_parameter] と [Parameter_value] を繰り返し表記することで可能となる。同時に、複数の種類の出力要素モデルや入力要素モデルの情報を変更した場合は、[Output_Directry_Change] や [Input_Directry_Change] を繰り返し表記することで可能となる。

[Calculate_end]

一回分の計算手順の終了を表す。

以下に例を示す。行頭の # マークはコメント行を表す記号である。

ソースコード 3: 計算手順指定ファイルの例

```
################################################################
[Calculate_start]
[Xml_Directry]
C:\home\tachikawa\CommonMP120\Execute\temp\test_storageFunctionProjectFile.xml
[Model_name_of_parameter]
KyotoUnivEngHywr.StorageFunctionMethod

[Parameter_value]
# <Variable Key="TIME_STEP" Name="タイムステップ（秒）" DataType="DOUBLE" Value="300" />
# <Variable Key="m_dArea" Name="流域面積␣A␣[km^2]" DataType="DOUBLE" Value="100" />
# <Variable Key="m_dK" Name="貯留係数␣K" DataType="DOUBLE" Value="30" />
# <Variable Key="m_dP" Name="貯留係数␣p" DataType="DOUBLE" Value="0.6" />
# <Variable Key="m_dTl" Name="遅滞時間␣TL␣[hr]" DataType="DOUBLE" Value="0" />
# <Variable Key="m_dF1" Name="1␣次流出率␣F1" DataType="DOUBLE" Value="0.7" />
# <Variable Key="m_dRsa" Name="飽和雨量␣Rsa␣[mm]" DataType="DOUBLE" Value="50" />
# <Variable Key="m_dR0" Name="初期損失雨量␣R0␣[mm]" DataType="DOUBLE" Value="0" />
# <Variable Key="m_dQb" Name="基底流量␣Qb␣[m^3/s]" DataType="DOUBLE" Value="10" />
# <Variable Key="m_dQ1" Name="初期流出高 (mm/hr)" DataType="DOUBLE" Value="0" />

300 120 30 0.6 0 0.7 50 0 20 10
```

```
[Output_Directry_Change]
#モデル名
Hymco.Prpr.StdMdl.WriteCSVFileModel

#ディレクトリー情報に関するKey
OverWriteFileNameWithPath

#新たなディレクトリー
.\OutputData\multiCulculate\case1\*

[Input_Directry_Change]
#モデル名
#ディレクトリー情報に関するKey
#新たなディレクトリー

[Calculate_end]

############################################################
[Calculate_start]
[Xml_Directry]
C:\home\tachikawa\CommonMP120\Execute\temp\test_storageFunctionProjectFile.xml
[Model_name_of_parameter]
KyotoUnivEngHywr.StorageFunctionMethod

[Parameter_value]
# <Variable Key="TIME_STEP" Name="タイムステップ（秒）" DataType="DOUBLE" Value="300" />
# <Variable Key="m_dArea" Name="流域面積␣A␣[km^2]" DataType="DOUBLE" Value="100" />
# <Variable Key="m_dK" Name="貯留係数␣K" DataType="DOUBLE" Value="30" />
# <Variable Key="m_dP" Name="貯留係数␣p" DataType="DOUBLE" Value="0.6" />
# <Variable Key="m_dTl" Name="遅滞時間␣TL␣[hr]" DataType="DOUBLE" Value="0" />
# <Variable Key="m_dF1" Name="1␣次流出率␣F1" DataType="DOUBLE" Value="0.7" />
# <Variable Key="m_dRsa" Name="飽和雨量␣Rsa␣[mm]" DataType="DOUBLE" Value="50" />
# <Variable Key="m_dR0" Name="初期損失雨量␣R0␣[mm]" DataType="DOUBLE" Value="0" />
# <Variable Key="m_dQb" Name="基底流量␣Qb␣[m^3/s]" DataType="DOUBLE" Value="10" />
# <Variable Key="m_dQ1" Name="初期流出高 (mm/hr)" DataType="DOUBLE" Value="0" />

300 150 50 0.6 0 0.7 50 0 10 10

[Output_Directry_Change]
#モデル名
Hymco.Prpr.StdMdl.WriteCSVFileModel

#ディレクトリー情報に関するKey
OverWriteFileNameWithPath

#新たなディレクトリー
.\OutputData\multiCulculate\case2\*


[Input_Directry_Change]
#モデル名
#ディレクトリー情報に関するKey
#新たなディレクトリー

[Calculate_end]
############################################################

}
```

6 パラメータ同定計算の手順

SCE-UA 法 (Shuffled Complex Evolution method) [3] を用いてパラメータ同定を行う。SCE-UA 法のプログラム SCE-UA.exe は、CMPEE の中で hymco.exe と交互によばれて、最適モデルパラメータを求める。ここで用いた SCE-UA.exe は多田によって Fortran90 を用いてコーディングされたプログラム [4] である。

SCE-UA.exe は CommonMP の実行プログラムをを置くフォルダ CommonMP/Execute/bin に置く。Cygwin で SCE-UA.exe を作成した場合は、cygwin のランタイムライブラリ cygwin1.dll も同じところにコピーする。

"Parameter_Identification" ラジオボタンにチェックを入れると**図 8** のようにパラメタ値の表の右側にパラメタ名が表記された列が新たに加わる。この時、新しく加わった列のセ

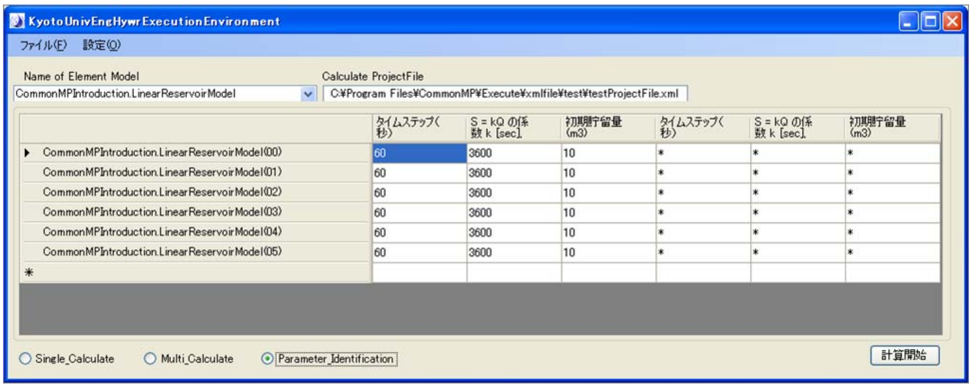


図 8: ”Parameter_Identification”ラジオボタンをチェックした後の画面

ルは全て「*」となっている。また本プログラムでは入力要素モデルと出力要素モデルはパラメタの同定を行えない仕様になっている。

この機能ではグループごとにパラメータの同定を行うことができる。**図 9** にグループの設定例を示す。同じグループには 0 以上の同じ整数を「*」に設定する。パラメータを変更しない場合は「*」のままにしておく。同じグループ内のパラメータ値は同じ値が常に代入されることになるため、異なるパラメータを同一グループに設定することはできない。

パラメタをグループ化し終えた後に”計算開始”ボタンを押すと、**図 10** のようにパラメタの各グループの値を設定する画面が開く。用いる整数の順番に意味はないが、連続した整数を用いる必要がある。**図 10** のように設定画面では以下の 5 項目を設定する必要がある。計算結果のデータファイルが存在しない場合は、”参照”ボタンから選択することが出来ないため、フルパスを直接入力する。

- 計算回数
- パラメタの初期値、最小値、最大値
- 比較する観測データと計算データのフルパス
- 目的関数
- データの時間表記 (Time Type)
- 計算結果のファイルへの出力の有無

目的関数は下記の 5 つを用意している。n : 計算時間数、Q_o : 実績流量、$Q_{o,peak}$: 実績ピーク流量、$T_{QMAX,O}$: 実績ピーク時刻、Q_s : 計算流量、$Q_{s,peak}$: 計算ピーク流量、$T_{QMAX,S}$: 計算ピーク時刻である。

Nash 指標 ハイドロ全体における各時刻の流量誤差を評価する。

$$Nash = 1 - \frac{\sum(Q_o - Q_s)^2}{\sum(Q_o - \bar{Q}_o)^2}$$

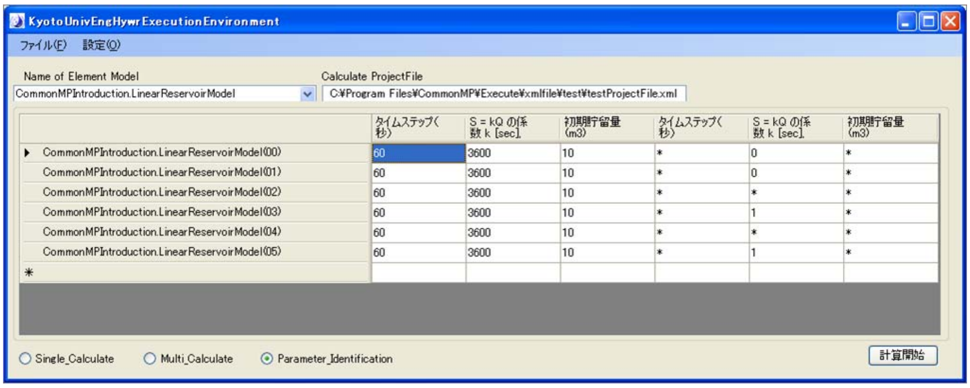


図 9: パラメータのグループ化

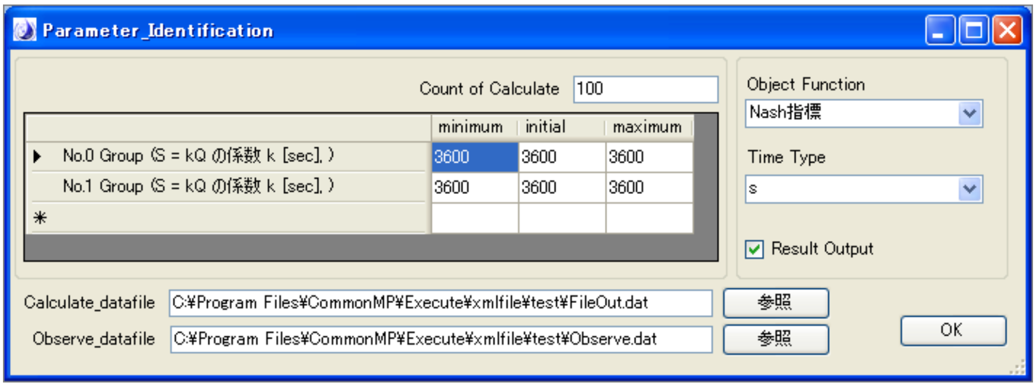


図 10: パラメータの値の範囲設定

流量誤差比率 ピーク流量に対する誤差比率を評価する。

$$E_Q = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{Q_o(i) - Q_s(i)}{Q_{o,peak}}\right)^2$$

ピーク流量誤差比率 ピーク時の流量比を評価する。

$$P_r = Q_{s,peak}/Q_{o,peak}$$

ピーク生起位相差 ピークの時間の是れを評価する。

$$TE = T_{QMAX,S} - T_{QMAX,O}$$

流出ボリューム誤差 流出ボリュームを評価する。

$$E_v = \frac{\sum Q_s - \sum Q_o}{\sum Q_o}$$

計算結果および観測データのデータ形式は**図 11** のようである。この形式は CommonMP の CSV 出力要素モデルを用いた場合の形式である。1 行目から 3 行目はヘッダで、4 行

```
HySCSVFileData,Ver1.0
データ区分, 時系列
Time, Data0
2012/09/07 15:00:00, 146.244454222388
2012/09/07 16:00:00, 243.845216742806
2012/09/07 17:00:00, 400.040071822879
⋮
```

図 11: 観測データと計算データのファイル形式

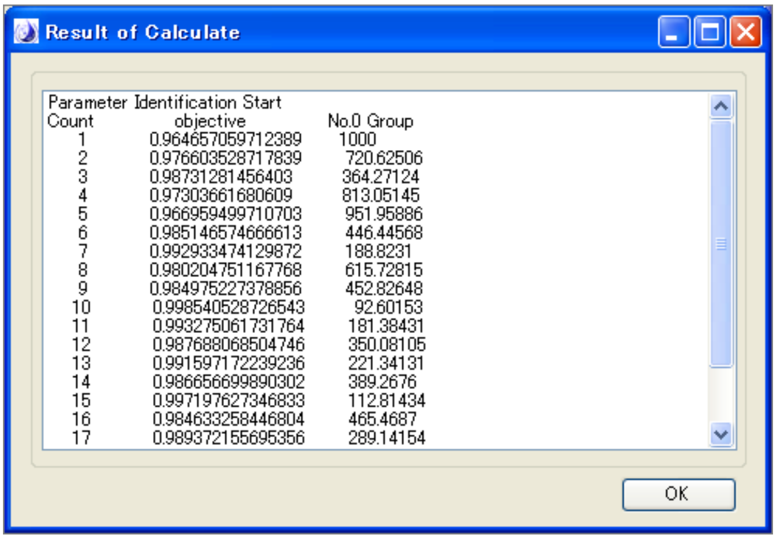


図 12: パラメータの値の同定の過程

目以降がデータとなる。4 行目以降の一つ目のカンマの次の数値を計算値あるいは観測値として読み込む。二つ目のカンマ以降にデータが記述されている場合は読み飛ばされる。現時点のプログラムでは、時間は見ていない。計算結果と観測データの始まりと終わりの時間は同じで、時間間隔も同じでなければばならない。

設定終了後 ”OK” ボタンを押すと計算が始まる。また同時に**図 12** のような画面が表示され計算番号、目的関数の値、パラメタ値が表示される。”Result Output” チェックボックスにチェックがある場合、計算終了後に**図 12** の内容が hymco.exe と同じディレクトリー下の ”cmpeeParamCalibResult.dat” に出力される。またこのファイルでは目的関数が一番良いものに「*」が付いている。

参考文献

[1] CommonMP, http://framework.nilim.go.jp/

[2] 高橋 円 : CommonMP の多重実行環境の開発と治水評価への適用, 京都大学修士学位論文, 2012.

[3] Qingyun Duan, Soroosh Sorooshian, and Vijai K. Gupta : Optimal use of the SCE-UA global optimization method for calibrating watershed models, Journal of Hydrology, 158, pp. 265–284, 1994.

[4] 多田 毅 : http://www.nda.ac.jp/cc/users/tada/