「品質工学セミナー」

付録 拡張直交表探索プログラム

2011.05.11. Y.Tanaka, K.Horino

１．Windows と Excel の組み合わせ

WindowsXP-Excel2003 ， WindowsXP-Excel2010

Windows7-Excel2010

本文書は pdf で提供されていますから、下記”**６．プログラムソース**”のプログラムを Excel の VBA エディターにコピーすればインストール完了です。プログラムはマクロ Main で起動します。

プログラムは上記組み合わせの Windows－Excel で走ることが確認されています。

【注】テキスト化では変換ミスが発生することがあります。原本 pdf を参照して修正してください。

２．使い方の簡単な説明

下記” **５．計算シート初期設定例**”を参考にシートに直交表他を記入して準備してください。

【注】文字”◆”は“しかく”または“きごう”を漢字変換すれば出てきます。

マクロ Main を実行すれば計算が開始されます。

最初に探索計算か解析計算かが問われます。

探索計算

直交表上で探索範囲（SearchRowStart, SearchRowEnd, SearchColStart, SearchColEnd）を指定し、次に探索カウントのスタート(cntUstart)とエンド(cntUend)を指定します。　計算が開始され、ステータスバーに探索回数(cntU)，合格回数(cntP)，探索途上の水準配列(vv)が表示されます。最後に「探索計算終了」のメッセージが表示されます。

探索は探索範囲（直交表で太線枠）のセルに次々水準値(1, 2, 3, ･･･)を代入し、できた直交表が解をもつかどうかを判定して行います。

探索結果は解リストに表示されます。ここで示した解リストは” **５．計算シート初期設定例**”に示されている L9+2 拡張直交表を探索した結果の１例です。

cntP は合格 No.、cntU は探索計算 No.、k は直交表の行番号。例えば探索範囲の行が 2 行(k=10 と k=11)で、列が 3 列であれば下図のように 3 桁の水準の配列（列番号順配列）が表示されます。

◆解リスト			
cntP	cntU	k = 10	k = 11
1	6	111	131
2	7	111	132
3	8	111	133
4	9	111	211
5	10	111	212
6	11	111	213
7	12	111	221
8	13	111	222
9	14	111	223

解析計算

直交表と実験値 y(k)を与え、因子効果、交互作用効果など(CalcVal)を計算します。計算が終了すれば、連立方程式、係数行列をシートに表示します。解けない直交表のときは CalcVal 行の次行に不合格と表示されます。

３．主なプロシージャの概略

【Main()】計算の根幹です。マクロ Main で起動すればプログラムが走ります。
【MitisuValSet()】未知数値（効果基準μ, 因子効果 a1, a2,･･･, 交互作用効果 ab1, ab2,･･･）の設定。
【YkCalc()】実験値 y(k)を、未知数値を用いて計算。
【KeisuMatrix()】係数行列生成。
【OrthoFacterKougoName()】主因子名、交互作用名の自動生成。
【OrthoTabKougo()】直交表の交互作用部作成。
【InitialSet()】直交表サイズなどの初期設定。
【LESQ_V2()】LINSW_V2()を用いて行列式を最小 2 乗法で解きます。
【LINSW_V2()】PivotH()を用いて掃き出し法で行列式を解きます。
【PivotH()】ピボット計算します。
【MakeEqs()】連立方程式を数式表現で作りシートに書き込みます。
【Hantei()】行列式の解があるか（True）ないか（False）を判定します。

４．その他

本プログラムは本書記載の計算を実施したものです。インストールや使用などにおいて不具合が生じても出版社および著者は責任を負えません。ユーザーの責任のもとに利用してください。プログラムの加工および転載を禁じます。

５．計算シート初期設定例

探索計算

	A	B	C	D	E	F	G
1					**KougoNum =**		1
2					**KougoName =**		abU
3	◆直交表		L9+2				
4		a	b	c	d	abU	y(k)
5		1	2	3	4	5	
6	k = 1	1	1	1	1	1	
7	2	1	2	2	2	2	
8	3	1	3	3	3	3	
9	4	2	2	3	1	3	
10	5	2	3	1	2	1	
11	6	2	1	2	3	2	
12	7	3	3	2	1	2	
13	8	3	1	3	2	3	
14	9	3	2	1	3	1	
15	10	2	1	1	1	2	
16	11	2	3	1	3	1	
17	Level	3	3	3	3	3	
18	◆探索条件						
19	**SearchRowStart**		10				
20	**SearchRowEnd**		11				
21	**SearchColStart**		2				
22	**SearchColEnd**		4				

解析計算

	A	B	C	D	E	F	G
1					**KougoNum =**		1
2					**KougoName =**		abU
3	◆直交表		L9+2				
4		a	b	c	d	abU	y(k)
5		1	2	3	4	5	
6	k = 1	1	1	1	1	1	
7	2	1	2	2	2	2	
8	3	1	3	3	3	3	
9	4	2	2	3	1	3	
10	5	2	3	1	2	1	
11	6	2	1	2	3	2	
12	7	3	3	2	1	2	
13	8	3	1	3	2	3	
14	9	3	2	1	3	1	
15	10	2	1	1	1	2	
16	11	2	3	1	3	1	
17	Level	3	3	3	3	3	
18	◆探索条件						
19	**SearchRowStart**		10				
20	**SearchRowEnd**		11				
21	**SearchColStart**		2				
22	**SearchColEnd**		4				

６．プログラムソース

```
Option Explicit
'==========================================================================
'Search32.09.xls (選択分岐は①探索、②解析の２ケースだけに変更)
'    Search of Expanded Orthogonal Table (拡張直交表探索)
'2011.05.13.Y.Tanaka, K.Horino
'==========================================================================
Const ESC = 0.0000000001
Const MaxN1 = 100 'max number of mitisu
Const KDF = 100# 'KougoDefaultFacter using as MitisuVal(mNr) = mNr / KDF
Const ScreenUpdate = False

'Q-break (push Q-key, you can break)←use Windows API
Private Declare Function GetAsyncKeyState _
                      Lib "User32.dll" (ByVal vKey As Long) As Long
'InitialSet
Dim SheetName As String, OrthoRowNum As Integer '
Dim rowNum As Integer, FacterNum As Integer, KougoNum As Integer
Dim rowStartXL As Double, colStartXL As Double
Dim SearchCondTop As Integer, OrthoTitleTop As Integer
'sweep out
Dim ResultLeft As Integer, ResultTop As Integer
Dim N As Integer, NRANK As Integer, NSTOP As Integer
'OrthoTab
Dim OrthoTab(MaxN1, MaxN1) As Double, LevelNum(MaxN1) As Integer
Dim OrthoTabName As String
Dim f As Integer, ff As Integer, k As Integer
Dim cntUstart As Long, cntUstop As Long, cntUbreak As Long
'MitisuSet
Dim MitisuMode As String, MitisuNr As Integer, MitisuNum As Integer
Dim MitisuVal(MaxN1) As Double, MitisuName(MaxN1) As String
Dim OrthoFacterName(MaxN1) As String, OrthoKougoName(MaxN1) As String
Dim SetMitisuTop As Integer, KougoNr As Integer
Dim M1 As Integer, N1 As Integer
'KeisuMatrix
Dim KeisuMatrixTop As Integer, KeisuMatrixLeft As Integer
Dim kMatrix(MaxN1, MaxN1) As Double
'ykCalc
Dim YkMode As String, y(MaxN1) As Double
'search
Dim SearchRowStart As Integer, SearchRowEnd As Integer
Dim SearchColStart As Integer, SearchColEnd As Integer
Dim LN As Integer
Dim c0 As String, c1 As String, c2 As String, c3 As String
Dim U As String, V As String
Dim vv As String
Dim SearchMode As Boolean
Dim cutColWidth As Integer, cutRowWidth As Integer
Dim V1(MaxN1) As String
Dim cntP As Long, cntU As Long 'パス（合格）カウント，サーチカウント
Dim SearchNum As Long
'specification of Kougo (解析交互作用の指定)
Dim SpecKougo(MaxN1) As String, SearchAreaLevelNum(MaxN1) As Integer
'MakeEqs (make the linear simultanious equations)
Dim EqsTop As Integer
Dim FacterName(20) As String
'ZeroSum(ゼロ和法)&AllCombExp (全組実験)
Dim AllCombExp As Boolean 'set at InitialSet() 全組合せ実験では探索しない
Dim ZeroEqsNum As Integer, OrthoEqsNum As Integer
'skip of sub YkCalc
```

```
Dim skipYkCalc As Boolean

Sub Main()
    Dim fNr As Integer, lineNr As Integer
    Dim i As Integer, ii As Integer, j As Integer
    Dim Maebu(61) As String, Atobu(61) As String
    Dim cEnd As String, msg As String, Row As Integer, Col As Integer
    Dim HanteiResult As Boolean, yesno As Integer, aa As String
    Dim B(MaxN1) As Double, BM(MaxN1) As Double
    Dim now0 As Variant, now1 As Variant
    Dim timer0 As Variant, timer1 As Variant, TimerSec As Long

'*****************************************************************************
MsgBox "ActiveSheet.Name = " & ActiveSheet.Name
'*****************************************************************************

    Application.StatusBar = ""
    cntU = 0 '【注】ないと入力未知数欄が表示されません
    cntP = 0
    'チェック(直交表仕上げまで先行)
    SearchMode = False '前回計算の SearchMode 値を消去
    InitialSet cntUstart, cntUstop
    OrthoTabKougo
    '探索／解析計算の選択
    yesno = MsgBox("SearchMode ?" & Chr(10) & " はい⇒探索計算" _
                        & Chr(10) & " いいえ⇒解析計算" _
                        & Chr(10) & " ｷｬﾝｾﾙ ⇒中止", vbYesNoCancel)
    Select Case yesno
    Case vbCancel
    Case vbNo
        Cells(1, 1).Value = "解析計算"
        YkMode = "HAND"
        'y(k)読み込みチェック
        For k = 1 To OrthoRowNum
            If Cells(k + rowStartXL - 1, _
                    FacterNum + KougoNum + 2).Value = "" Then
                MsgBox "直交表右端列の y(k)値が記入されてません"
                End
            End If
        Next k
        MitisuMode = "HAND"
        For k = 1 To OrthoRowNum
            y(k) = Val(Cells(k + rowStartXL - 1, _
                        FacterNum + KougoNum + 2).Value)
        Next k
    Case vbYes
        Cells(1, 1).Value = "探索計算"
        YkMode = "AUTO"
        skipYkCalc = True
        MitisuMode = "AUTO"
    Case Else
        MsgBox "error in Main() No.4"
        End
    End Select
    Select Case yesno
    Case vbCancel
        End
    Case vbNo
        '単発計算
        Erase OrthoTab
```

```
    SearchMode = False
    cntU = 0 'ないと入力未知数欄が表示されません
    InitialSet cntUstart, cntUstop
    OrthoTabKougo
    MitisuValSet
    SetMitisu
    KeisuMatrix
    YkCalc
    GoSub LESQcalc
    If Hantei = True Then
        msg = "合格 NRANK =" & Str(NRANK) & " NSTOP =" & Str(NSTOP)
    Else
        msg = "不合格 NRANK =" & Str(NRANK) & " NSTOP =" & Str(NSTOP)
    End If
    Cells(SetMitisuTop + 4, 3).Value = msg
    Cells(SetMitisuTop + 4, 3).HorizontalAlignment = xlLeft
    Cells(SetMitisuTop + 4, 3).Select
    MakeEqs
    MsgBox "計算終了  " & msg
    End
Case vbYes
    SearchMode = True
    '探索計算開始
    SearchNum = 1
    For Row = SearchRowStart To SearchRowEnd
        For Col = SearchColStart To SearchColEnd
            SearchNum = SearchNum * LevelNum(Col)
        Next Col
    Next Row
    Cells(SearchCondTop + 5, 1).Value = "SearchNum =" & Str(SearchNum)
    Cells(SearchCondTop + 5, 1).HorizontalAlignment = xlLeft
    Cells(SearchCondTop + 6, 3).Select
    cntUstart = Val(InputBox("スタート・ランNo. cntU = ?", , "0"))
    cntUstop = Val(InputBox("エンド・ランNo. cntUstop = ?", , "0"))
    If SearchNum < cntUstop Then
        cntUstop = SearchNum
    End If
    InitialSet cntUstart, cntUstop
    GoSub time1
    '最初は直交表の表示を仕上げます
    OrthoTabKougo
    DispSearchArea
    MitisuValSet
    YkCalc
    SetMitisu
    KeisuMatrix
    '画面更新の停止
    Application.ScreenUpdating = ScreenUpdate
    '解リスト（モニター）項目行
    cntU = 0
    If cntU = 0 Then
        If SearchMode = True Then
            Cells(ResultTop, 1).Value = "◆解リスト"
            Cells(ResultTop, 1).HorizontalAlignment = xlLeft
            Cells(ResultTop + 1, 1).Value = "cntP"
            Cells(ResultTop + 1, 2).Value = "cntU"
            For k = SearchRowStart To SearchRowEnd
                Cells(ResultTop + 1, k - SearchRowStart + 3).Value _
                                                = "k =" & Str(k)
            Next k
```

```
        For MitisuNr = 1 To MitisuNum
            Cells(ResultTop + 1, MitisuNr + cutRowWidth + 2).Value _
                                        = MitisuName(MitisuNr)
        Next MitisuNr
    End If
End If
'網羅的探索
cntU = cntUstart
cntP = 0
If cntU < 0 Then
    cntU = 0
End If
U = Change10toLN(cntU, LN)
If SearchNum < cntUstop Then
    cntUstop = SearchNum
End If
Do
    U = Change10toLN(cntU, LN)
    vv = ChangeUtoV(U) 'vv は探索領域シリアル
    For lineNr = SearchRowStart To SearchRowEnd
        ii = lineNr - SearchRowStart + 1
        i = cutColWidth * (ii - 1) + 1
        'vv を探索部 1 行ずつに分けます
        V1(ii) = Mid(vv, i, cutColWidth)
        'OrthoTab 更新
        For fNr = SearchColStart To SearchColEnd
            OrthoTab(lineNr, fNr) = Val(Mid(V1(ii), _
                    fNr - SearchColStart + 1, 1))
        Next fNr
    Next lineNr
    OrthoTabKougo
    DispOrthoTab
    YkCalc
    KeisuMatrix
    GoSub LESQcalc
    '解の判定
    If Hantei = True And cntU >= 0 Then
        cntP = cntP + 1
        DispResult cntP, cntU
    End If
    Application.StatusBar = "cntU =" & Str(cntU) & _
                        ", cntP =" & Str(cntP) & ", vv = " & vv
    cntU = cntU + 1 '次回の準備
    '"Q"キー(Quit)強制中断の処置
    If GetAsyncKeyState(vbKeyQ) Then
        Application.ScreenUpdating = True
        U = Change10toLN(cntU, LN)
        vv = ChangeUtoV(U) 'vv は探索領域シリアル
        GoSub BreakCalc
        If MsgBox("OK:終了, ｷｬﾝｾﾙ:継続", vbOKCancel) = vbOK Then
            Exit Do
        End If
        Application.ScreenUpdating = ScreenUpdate
    End If
Loop Until cntU > cntUstop
cntU = cntU - 1
Cells(SearchCondTop + 6, 1).Value = "cntUstart/stop =" _
                            & Str(cntUstart) & " ～" & Str(cntU)
Cells(SearchCondTop + 6, 1).HorizontalAlignment = xlLeft
U = Change10toLN(cntU, LN)
```

```
        vv = ChangeUtoV(U) 'vvは探索領域シリアル
        GoSub BreakCalc
        If SearchMode = True Then
            GoSub time2
        End If
        '画面更新の復活
        Application.ScreenUpdating = True
        MsgBox "探索計算終了"
        Application.StatusBar = ""
    Case Else
        MsgBox "error in Main() No.4"
        End
    End Select
    End
Exit Sub

BreakCalc:
    '最終y(k)と直交表を記入（探索領域行のみ）
    OrthoTabKougo
    DispOrthoTab
    YkCalc
    If skipYkCalc = False Then
        For lineNr = SearchRowStart To SearchRowEnd
            Cells(lineNr + rowStartXL - 1, _
                    FacterNum + KougoNum + 2).Value = y(lineNr)
        Next lineNr
    End If
    cntUbreak = cntU
    Cells(SearchCondTop + 5, 6).Value = "cntP =" & Str(cntP)
    Cells(SearchCondTop + 5, 6).HorizontalAlignment = xlLeft
    Cells(SearchCondTop + 6, 6).Value = "Hit% =" & Str(Int(cntP _
            / (cntUbreak - cntUstart + 1) * 100# * 100#)) / 100# & "%"
    Cells(SearchCondTop + 6, 6).HorizontalAlignment = xlLeft
    GoSub time2
Return

time1:
    now0 = Now
    timer0 = Timer
    Cells(SearchCondTop + 1, 6).Value = "START"
    Cells(SearchCondTop + 1, 6).HorizontalAlignment = xlLeft
    Cells(SearchCondTop + 1, 7).Value = " " & now0
    Cells(SearchCondTop + 1, 7).HorizontalAlignment = xlLeft
Return

time2:
    now1 = Now
    timer1 = Timer
    TimerSec = (timer1 - timer0) + Int(now1 - now0) * 86400
    Cells(SearchCondTop + 2, 6).Value = "END"
    Cells(SearchCondTop + 2, 6).HorizontalAlignment = xlLeft
    Cells(SearchCondTop + 2, 7).Value = " " & Now
    Cells(SearchCondTop + 2, 7).HorizontalAlignment = xlLeft
    '所用時間
    TimerSec = (timer1 - timer0) + Int(now1 - now0) * 86400
    Cells(SearchCondTop + 3, 6).Value = "TimerSec =" & Str(TimerSec)
    Cells(SearchCondTop + 3, 6).HorizontalAlignment = xlLeft
    Cells(SearchCondTop + 4, 6).Value = "Velocity"
    Cells(SearchCondTop + 4, 7).Value = Str(1000 * CDbl(TimerSec) _
                        / CDbl(cntU - cntUstart + 1))
```

```
    Cells(SearchCondTop + 4, 7).HorizontalAlignment = xlCenter
    Cells(SearchCondTop + 4, 8).NumberFormatLocal = "0.0_ "
    Cells(SearchCondTop + 4, 8).Value = "msec／run"
    Cells(SearchCondTop + 4, 8).HorizontalAlignment = xlLeft
Return

LESQcalc:
    For k = 1 To rowNum
        BM(k) = y(k)
    Next k
    NSTOP = 0
    LESQ_V2 kMatrix, BM, B, rowNum, MitisuNum, NSTOP
    For MitisuNr = 1 To MitisuNum
        kMatrix(MitisuNr, MitisuNum + 1) = B(MitisuNr)
        Cells(SetMitisuTop + 3, MitisuNr + 1).Value = B(MitisuNr)
        Cells(SetMitisuTop + 3, MitisuNr + 1).HorizontalAlignment = xlCenter
    Next MitisuNr
Return
End Sub

Sub SetMitisu()
'未知数設定
    Dim j As Integer

    Cells(SetMitisuTop, 1).Value = "◆未知数（効果基準, 因子効果,他）"
    Cells(SetMitisuTop, 1).HorizontalAlignment = xlLeft
    For j = 1 To MitisuNum
        Cells(SetMitisuTop + 1, j + 1).Value = MitisuName(j)
        If YkMode = "AUTO" Then
            Cells(SetMitisuTop + 2, j + 1).Value = MitisuVal(j)
        Else
            Cells(SetMitisuTop + 2, j + 1).Value = "･･･"
        End If
    Next j

    If MitisuMode = "AUTO" Then
        Cells(SetMitisuTop + 2, 1).Value = "AutoVal(仮設定)"
    Else
        Cells(SetMitisuTop + 2, 1).Value = "HandVal"
    End If
    Cells(SetMitisuTop + 2, 1).HorizontalAlignment = xlLeft
    Cells(SetMitisuTop + 3, 1).Value = "CalcVal(確認)"
    Cells(SetMitisuTop + 3, 1).HorizontalAlignment = xlLeft

End Sub

Sub MitisuValSet()
    Dim Level As Integer, s As Integer, j As Integer, sum As Double
    Dim EqsNum As Integer, Margin As Integer
    Dim mNr As Integer 'MitisuNr の置き換え

    mNr = 1
    MitisuName(mNr) = "μ" '←μは mNr = 1
    For f = 1 To FacterNum
        For Level = 1 To LevelNum(f)
            mNr = mNr + 1
            MitisuName(mNr) = OrthoFacterName(f) & LTrim(Str(Level))
        Next Level
    Next f
    If KougoNum > 0 Then
```

```
    For KougoNr = 1 To KougoNum
        For Level = 1 To LevelNum(FacterNum + KougoNr)
            mNr = mNr + 1
            MitisuName(mNr) = OrthoKougoName(KougoNr) _
                                        & LTrim(Str(Level))
        Next Level
    Next KougoNr
End If
MitisuNum = mNr
EqsNum = OrthoRowNum
ZeroEqsNum = FacterNum + KougoNum
Cells(SearchCondTop + 1, 11).Value = "EqsNum =" & Str(EqsNum)
Cells(SearchCondTop + 1, 11).HorizontalAlignment = xlLeft
Cells(SearchCondTop + 2, 11).Value = "ZeroEqsNum =" & Str(ZeroEqsNum)
Cells(SearchCondTop + 2, 11).HorizontalAlignment = xlLeft
Cells(SearchCondTop + 3, 11).Value = "MitisuNum =" & Str(MitisuNum)
Cells(SearchCondTop + 3, 11).HorizontalAlignment = xlLeft
Margin = EqsNum + ZeroEqsNum - MitisuNum
Cells(SearchCondTop + 4, 11).Value = "Margin =" & Str(EqsNum) & "+" & _
    LTrim(Str(ZeroEqsNum)) & "-" & LTrim(Str(MitisuNum)) _
    & " =" & Str(Margin)
Cells(SearchCondTop + 4, 11).HorizontalAlignment = xlLeft
Cells(SearchCondTop + 5, 11).Value = "AllCombExp = " & AllCombExp
Cells(SearchCondTop + 5, 11).HorizontalAlignment = xlLeft
If Margin < 0 Then
    MsgBox "式数が未知数個数より少ないので計算を打ち切ります。"
    End
End If
Cells(SearchCondTop + 6, 11).Value = ThisWorkbook.Name
Cells(SearchCondTop + 6, 11).HorizontalAlignment = xlLeft
'係数行列サイズ
M1 = rowNum
N1 = MitisuNum
If YkMode = "HAND" Then 'calculated without relation to mitisu value
    Exit Sub
End If
'未知数 HAND 設定
If MitisuMode = "HAND" Then
    Cells(3, 5).Value = "MitisuName ="
    Cells(3, 5).Font.Bold = True
    Cells(3, 5).HorizontalAlignment = xlLeft
    Cells(4, 5).Value = "HandSetVal ="
    Cells(4, 5).Font.Bold = True
    Cells(4, 5).HorizontalAlignment = xlLeft
    For MitisuNr = 1 To MitisuNum
        Cells(3, 6 + MitisuNr).Value = MitisuName(MitisuNr)
        Cells(3, 6 + MitisuNr).HorizontalAlignment = xlCenter
        If Len(Cells(4, 6 + MitisuNr).Value) = 0 Then
            Cells(4, 6 + MitisuNr).Value = 0
        End If
        MitisuVal(MitisuNr) = Val(Cells(4, 6 + MitisuNr).Value)
    Next MitisuNr
    '未知数未入力チェック
    For MitisuNr = 2 To MitisuNum
        sum = sum + MitisuVal(MitisuNr) ^ 2
    Next MitisuNr
    If sum < ESC Then
        MsgBox "第 4 行で設定未知数(HandSetVal)の値が記入されていません"
        End
    End If
```

```
ElseIf MitisuMode = "AUTO" Then
    '主効果値の自動設定
    MitisuVal(1) = 10
    mNr = 1
    For f = 1 To FacterNum
        For Level = 1 To LevelNum(f)
            mNr = mNr + 1
            Select Case LN
            Case 0
                Select Case LevelNum(f)
                Case 2
                    Select Case Level
                    Case 1
                        MitisuVal(mNr) = f
                    Case 2
                        MitisuVal(mNr) = -MitisuVal(mNr - 1)
                    Case Else
                        MsgBox "Fatal error in MitisuValSet() No.1"
                    End Select
                Case 3
                    Select Case Level
                    Case 1
                        MitisuVal(mNr) = f
                    Case 2
                        MitisuVal(mNr) = MitisuVal(mNr - 1) + 0.5
                    Case 3
                        MitisuVal(mNr) = -MitisuVal(mNr - 2) - MitisuVal(mNr - 1)
                    Case Else
                    End Select
                Case 4
                    Select Case Level
                    Case 1
                        MitisuVal(mNr) = f
                    Case 2
                        MitisuVal(mNr) = MitisuVal(mNr - 1) / 2#
                    Case 3
                        MitisuVal(mNr) = MitisuVal(mNr - 1) / 4#
                    Case 4
                        MitisuVal(mNr) = -MitisuVal(mNr - 1) _
                                    - MitisuVal(mNr - 2) - MitisuVal(mNr - 3)
                    Case Else
                        MsgBox "Fatal error in MitisuValSet() No.2"
                    End Select
                Case Else
                    MsgBox "Fatal error in MitisuValSet() No.3"
                End Select
            Case 2
                Select Case Level
                Case 1
                    MitisuVal(mNr) = f
                Case 2
                    MitisuVal(mNr) = -MitisuVal(mNr - 1)
                Case Else
                    MsgBox "Fatal error in MitisuValSet() No.4"
                End Select
            Case 3
                Select Case Level
                Case 1
                    MitisuVal(mNr) = f
                Case 2
```

```
                MitisuVal(mNr) = MitisuVal(mNr - 1) + 0.5
            Case 3
                MitisuVal(mNr) = -MitisuVal(mNr - 2) - MitisuVal(mNr - 1)
            Case Else
            End Select
        Case Else
            MsgBox "Fatal error in MitisuValSet() No.5"
        End Select
    Next Level
Next f
'交互作用値の自動設定
If KougoNum > 0 Then
    For KougoNr = 1 To KougoNum
        For Level = 1 To LevelNum(FacterNum + KougoNr)
            mNr = mNr + 1
            Select Case LN
            Case 0
                Select Case LevelNum(FacterNum + KougoNr)
                Case 2
                    Select Case Level
                    Case 1
                       MitisuVal(mNr) = CSng(mNr) / KDF
                    Case 2
                        MitisuVal(mNr) = -MitisuVal(mNr - 1)
                    Case Else
                        MsgBox "Fatal error in MitisuValSet() No.6"
                    End Select
                Case 3
                    Select Case Level
                    Case 1
                        MitisuVal(mNr) = CSng(mNr) / KDF
                    Case 2
                        MitisuVal(mNr) = MitisuVal(mNr - 1) + 0.0005
                    Case 3
                        MitisuVal(mNr) = -MitisuVal(mNr - 2) - MitisuVal(mNr - 1)
                    Case Else
                    End Select
                Case 4
                    Select Case Level
                    Case 1
                        MitisuVal(mNr) = CSng(mNr) / KDF
                    Case 2
                        MitisuVal(mNr) = MitisuVal(mNr - 1) / 2#
                    Case 3
                        MitisuVal(mNr) = MitisuVal(mNr - 1) / 4#
                    Case 4
                        MitisuVal(mNr) = -MitisuVal(mNr - 1) _
                            - MitisuVal(mNr - 2) - MitisuVal(mNr - 3)
                    Case Else
                        MsgBox "Fatal error in MitisuValSet() No.7"
                    End Select
                Case Else
                    MsgBox "Fatal error in MitisuValSet() No.8"
                End Select
            Case 2
                Select Case Level
                Case 1
                    MitisuVal(mNr) = mNr / KDF
                Case 2
                    MitisuVal(mNr) = -MitisuVal(mNr - 1)
```

```
                    Case Else
                        MsgBox "Fatal error in MitisuValSet() No.9"
                    End Select
                Case 3
                    Select Case Level
                    Case 1
                        MitisuVal(mNr) = mNr / KDF
                    Case 2
                        MitisuVal(mNr) = _
                                MitisuVal(mNr - 1) + 0.1 / 100#
                    Case 3
                    MitisuVal(mNr) = -MitisuVal(mNr - 2) - MitisuVal(mNr - 1)
                    Case Else
                    End Select
                Case 4
                    Select Case Level
                    Case 1
                        MitisuVal(mNr) = mNr / KDF
                    Case 2
                        MitisuVal(mNr) = mNr / KDF * (1# + 0.5)
                    Case 3
                        MitisuVal(mNr) = -MitisuVal(mNr - 1)
                    Case 4
                        MitisuVal(mNr) = -MitisuVal(mNr - 1) _
                                - MitisuVal(mNr - 2) - MitisuVal(mNr - 3)
                    Case Else
                        MsgBox "Fatal error in MitisuValSet() No.10"
                    End Select
                Case Else
                    MsgBox "Fatal error in MitisuValSet() No.11"
                End Select
            Next Level
        Next KougoNr
    End If
Else
    MsgBox "Error in MitisuValSet() No.12"
End If
For mNr = 2 To MitisuNum
    sum = sum + MitisuVal(mNr) ^ 2
Next mNr
If sum = 0 Then
    MsgBox "設定未知数の値が記入されていません"
    End
End If
End Sub

Sub YkCalc()
'y(k)デフォルト値の算出
    Dim Row As Integer, x As Double
    Dim j As Integer, jStart(100) As Integer

    If YkMode = "HAND" Then
        Exit Sub
    End If
    '未知数のスタート位置 MitisuNr
    jStart(1) = 2
    For j = 2 To FacterNum + KougoNum
        jStart(j) = jStart(j - 1) + LevelNum(j - 1)
    Next j
    'y(k)を計算します
```

```
        For k = 1 To OrthoRowNum
            y(k) = MitisuVal(1)
            For j = 1 To FacterNum + KougoNum
                Select Case OrthoTab(k, j)
                Case 1: x = MitisuVal(jStart(j))
                Case 2: x = MitisuVal(jStart(j) + 1)
                Case 3: x = MitisuVal(jStart(j) + 2)
                Case 4: x = MitisuVal(jStart(j) + 3)
                Case -1: x = -MitisuVal(jStart(j))
                Case -2: x = -MitisuVal(jStart(j) + 1)
                Case -3: x = -MitisuVal(jStart(j) + 2)
                Case -4: x = -MitisuVal(jStart(j) + 3)
                Case Else
                    MsgBox "Fatal error in YkCalc() No.1"
                End Select
                y(k) = y(k) + x
            Next j
            If skipYkCalc = False Then
                '直交表右端列に記入
                Cells(k + rowStartXL - 1, FacterNum + KougoNum + 2).Value = y(k)
            End If
        Next k
    For k = OrthoRowNum + 1 To M1
        y(k) = 0#
    Next k
    For k = 1 To M1
        '係数行列右端列に記入
        kMatrix(k, M1 + 1) = y(k)
    Next k
End Sub

Sub KeisuMatrix()
    Dim KeisuMatrixYoko(MaxN1) As String
    Dim j As Integer, jj As Integer, jjj As Integer, jjNum As Integer
    Dim KougoCnt As Integer, L As Integer, cnt As Integer

    Erase kMatrix
    KeisuMatrixLeft = colStartXL
    KeisuMatrixTop = SetMitisuTop + 10
    '最初全要素を0にする
    For k = 1 To M1
        For j = 1 To N1
            kMatrix(k, j) = 0
        Next j
    Next k
    '係数行列記入(1)：μ列
    For k = 1 To OrthoRowNum
        kMatrix(k, 1) = 1
    Next k
    '係数行列記入(2)：主効果
    For k = 1 To OrthoRowNum
        jj = 0
        For j = 1 To FacterNum
            jj = jj + LevelNum(j - 1)
            jjj = jj + OrthoTab(k, j) + 2 - 1
            kMatrix(k, jjj) = 1
        Next j
    Next k
    jjNum = jj '主効果の未知数個数
    '係数行列記入(3)：U&V交互作用列
```

```
    For k = 1 To OrthoRowNum
        jj = jjNum
        For j = FacterNum + 1 To FacterNum + KougoNum
            jj = jj + LevelNum(j - 1)
            jjj = jj + Abs(OrthoTab(k, j)) + 2 - 1
            If OrthoTab(k, j) > 0 Then
                kMatrix(k, jjj) = 1
            ElseIf OrthoTab(k, j) < 0 Then
                kMatrix(k, jjj) = -1
            End If
        Next j
    Next k
    '係数行列記入(4)：ゼロ和条件式
    j = 0
    For k = OrthoRowNum + 1 To OrthoRowNum + FacterNum + KougoNum
        cnt = k - OrthoRowNum
        j = j + LevelNum(cnt)
        For L = 1 To LevelNum(cnt)
            Select Case LevelNum(cnt)
            Case 2
                kMatrix(k, j + L - 1) = 1
            Case 3
                kMatrix(k, j + L - 2) = 1
            Case 4
                kMatrix(k, j + L - 3) = 1
            Case Else
                MsgBox "Fatal error in KeisuMatrix() No.1"
            End Select
        Next L
    Next k
    rowNum = OrthoRowNum + cnt
    If SearchMode <> True Then
        DispKeisuMatrix
    End If
End Sub

Sub DispKeisuMatrix()
    Dim j As Integer, jj As Integer, k As Integer

    '係数行列タテヨコ項目記入
    Cells(KeisuMatrixTop - 2, 1).Value = "◆係数行列 kMatrix(i,j)"
    Cells(KeisuMatrixTop - 2, 1).HorizontalAlignment = xlLeft
    For j = 1 To MitisuNum 'M1
        Cells(KeisuMatrixTop - 1, j + KeisuMatrixLeft - 1).Value _
                                                = MitisuName(j)
    Next j
    For k = 1 To rowNum 'M1
        Cells(k + KeisuMatrixTop - 1, 1).Value = k
    Next k
    Cells(KeisuMatrixTop - 2, 5).Value = "M1 =" & Str(MitisuNum) & _
                                    "行, N1=" & Str(rowNum) & "列"
    Cells(KeisuMatrixTop - 2, 5).HorizontalAlignment = xlLeft
    Range(Cells(KeisuMatrixTop - 1, KeisuMatrixLeft - 1), _
                                    Cells(KeisuMatrixTop + M1 - 1, _
                                    KeisuMatrixLeft + N1 - 1)).Select
    Selection.HorizontalAlignment = xlCenter
    '全要素を表示(右端 y(k)は別箇所 YkCalc で表示)
    For k = 1 To rowNum 'M1
        For j = 1 To N1
            Cells(k + KeisuMatrixTop - 1, _
```

```
                            KeisuMatrixLeft + j - 1).Value = kMatrix(k, j)
        Next j
    Next k
    Cells(KeisuMatrixTop + MitisuNum, 1).Select
End Sub

Sub OrthoFacterAndKougoName()
    Dim Level As Integer, KougoNr As Integer, cntF As Integer

    '主効果名
    cntF = 0
    For f = 1 To FacterNum
        OrthoFacterName(f) = Chr(96 + f)
        Cells(rowStartXL - 2, colStartXL + f - 1).Value _
                                        = OrthoFacterName(f)
        cntF = cntF + 1
        Cells(rowStartXL - 1, colStartXL + cntF - 1).Value = cntF
    Next f
    '交互作用名
    For KougoNr = 1 To KougoNum
        OrthoKougoName(KougoNr) = SpecKougo(KougoNr)
        Cells(rowStartXL - 2, colStartXL + FacterNum + KougoNr - 1).Value _
                                            = OrthoKougoName(KougoNr)
        Cells(rowStartXL - 1, colStartXL + cntF - 1).Value = cntF
    Next KougoNr
End Sub

Sub OrthoTabKougo()
'交互作用部作成(Kougo: interaction of orthogonal table)
    Dim kougo(5, 5) As Integer, kougoU(5, 5) As Integer
    Dim ss As Integer, cntF As Integer
    Dim c As String, LargerLevel As String, SmallerLevel As String
    Dim g As Integer, j As Integer, Col As Integer
    Dim row1 As Integer, col1 As Integer, row2 As Integer, col2 As Integer

    If cntU = 0 Then
        '未知数名記入
        OrthoFacterAndKougoName
    End If
    '交互作用部作成
    For k = 1 To OrthoRowNum
        For KougoNr = 1 To KougoNum
            f = Asc(Mid(SpecKougo(KougoNr), 1, 1)) - 96
            ff = Asc(Mid(SpecKougo(KougoNr), 2, 1)) - 96
            If f < 0 Or ff < 0 Then
                MsgBox "交互作用名 " & SpecKougo(KougoNr) _
                                        & " を半角にしてください"
                End
            End If
            If LevelNum(f) > LevelNum(ff) Then
                LargerLevel = LTrim(Str(LevelNum(f)))
                SmallerLevel = LTrim(Str(LevelNum(ff)))
            Else
                LargerLevel = LTrim(Str(LevelNum(ff)))
                SmallerLevel = LTrim(Str(LevelNum(f)))
            End If
            c = SmallerLevel & "×" & LargerLevel
            GoSub kougoDef
            '交互作用部記入
            j = FacterNum + KougoNr
```

```
            OrthoTab(k, j) = kougo(OrthoTab(k, f), OrthoTab(k, ff))
            LevelNum(j) = Val(LargerLevel)
            If cntU = 0 Then
                Cells(k + rowStartXL - 1, KougoNr + FacterNum + 1).Value = _
                                    kougo(OrthoTab(k, f), OrthoTab(k, ff))
                Cells(OrthoRowNum + rowStartXL, KougoNr + _
                                    FacterNum + 1).Value = LevelNum(j)
            End If
            If cntU = 0 Then
                '交互作用部のｆ番号
                Cells(rowStartXL - 1, FacterNum + KougoNr + 1).Value _
                                          = FacterNum + KougoNr
            End If
        Next KougoNr
    Next k
    '直交表右端へ"y(k)"記入
    Cells(rowStartXL - 2, colStartXL + FacterNum _
                                    + KougoNum).Value = "y(k)"
    If cntU = cntUstart Then
        '交互作用部罫線
        If KougoNum > 0 Then
            row1 = rowStartXL
            row2 = row1 + OrthoRowNum - 1
            col1 = colStartXL + FacterNum
            col2 = col1 + KougoNum - 1
            Keisen row1, col1, row2, col2
        End If
    End If
    Cells(SearchCondTop + 5, 1).Select
Exit Sub

kougoDef: '交互作用定義
    Select Case c
    Case "2×2"
        Select Case Right(SpecKougo(KougoNr), 1)
        Case "U", "V"
            MsgBox "交互作用名末尾に不要のＵまたはＶがあります"
            End
        Case Else
            kougo(1, 1) = 1
            kougo(1, 2) = 2
            kougo(2, 1) = 2
            kougo(2, 2) = 1
        End Select
    Case "2×3"
        If Right(SpecKougo(KougoNr), 1) <> "W" Then
            MsgBox "KougoName の末尾""W""をつけなさい"
            End
        End If
        kougo(1, 1) = 1
        kougo(1, 2) = 2
        kougo(1, 3) = 3
        kougo(2, 1) = -1
        kougo(2, 2) = -2
        kougo(2, 3) = -3
    Case "2×4"
        If Right(SpecKougo(KougoNr), 1) <> "Z" Then
            MsgBox "KougoName の末尾""Z""をつけなさい"
            End
        End If
```

```
        kougo(1, 1) = 1
        kougo(2, 1) = 2
        kougo(3, 1) = 3
        kougo(4, 1) = 4
        kougo(1, 2) = -1
        kougo(2, 2) = -2
        kougo(3, 2) = -3
        kougo(4, 2) = -4
    Case "3×3"
        Select Case Right(SpecKougo(KougoNr), 1)
        Case "U"
            kougo(1, 1) = 1
            kougo(1, 2) = 2
            kougo(1, 3) = 3
            kougo(2, 1) = 2
            kougo(2, 2) = 3
            kougo(2, 3) = 1
            kougo(3, 1) = 3
            kougo(3, 2) = 1
            kougo(3, 3) = 2
        Case "V"
            kougo(1, 1) = 3
            kougo(1, 2) = 2
            kougo(1, 3) = 1
            kougo(2, 1) = 1
            kougo(2, 2) = 3
            kougo(2, 3) = 2
            kougo(3, 1) = 2
            kougo(3, 2) = 1
            kougo(3, 3) = 3
        Case Else
            MsgBox "交互作用名末尾でUまたはVが欠落しています"
            End
        End Select
    Case "2×4"
        If Right(SpecKougo(KougoNr), 1) <> "W" Then
            MsgBox "KougoName の末尾""W""をつけなさい"
            End
        End If
        kougo(1, 1) = 1
        kougo(2, 1) = -1
        kougo(1, 2) = 2
        kougo(2, 2) = -2
        kougo(1, 3) = 3
        kougo(2, 3) = -3
        kougo(1, 4) = 4
        kougo(2, 4) = -4
    Case Else
        MsgBox "Fatal error in OrthoTabKougo() No.1"
        End
    End Select
Return

End Sub

Private Function ChangeUtoV(ByVal U As String)
'純粋LN進数(0,1,2,...,(n-1))を直交表表記(1,2,3,...,n)に変換します
'LN<=9の場合のみ有効です
    Dim A(100) As Integer
    Dim keta As Integer, i As Integer
```

```
    keta = Len(U)
    V = ""
    For i = 1 To keta
        A(i) = Val(Mid(U, i, 1)) + 1
        V = V & LTrim(Str(A(i)))
    Next i
    ChangeUtoV = V
End Function

Function ChangeVtoU(ByVal V As String) As String
'直交表表記(1,2,3,...,n)を純粋LN進数(0,1,2,...,(n-1))に変換します
'LN<=9の場合のみ有効です
    Dim A(100) As Integer
    Dim keta As Integer, i As Integer

    keta = Len(V)
    U = ""
    For i = 1 To keta
        A(i) = Val(Mid(V, i, 1)) - 1
        U = U & LTrim(Str(A(i)))
    Next i
    ChangeVtoU = U
End Function

Function Change10toLN(ByVal x As Long, LN As Integer) As String
'10進数xをLN進数に変換して、keta桁文字列(0,1,2)として返します
    Dim t As Long, r As String, LNN As Integer
    Dim keta As Integer, i As Integer, Row As Integer, Col As Integer

    If x = 0 Then Change10toLN = c0: Exit Function
    '【注】探索3行3列のとき9桁
    t = 1&
    r = ""
    If LN <> 0 Then
        Do While (t <= x)
            r = CStr((x Mod (t * LN)) ¥ t) & r
            t = t * LN
        Loop
        keta = Len(r)
    Else
        For Row = SearchRowEnd To SearchRowStart Step -1
            For Col = SearchColEnd To SearchColStart Step -1
                LNN = LevelNum(Col) 'Colは既に与えられている
                r = CStr((x Mod (t * LNN)) ¥ t) & r
                t = t * LNN
            Next Col
        Next Row
    End If
    keta = Len(r)
    If keta < Len(c0) Then
        For i = keta + 1 To Len(c0)
            r = "0" & r
        Next i
    End If
    Change10toLN = r
End Function

Sub InitialSet(cntUstart As Long, cntUstop As Long)
'直交表主効果部読込み
```

```
Dim i As Integer, j As Integer, k As Integer
Dim SearchCellNum As Long, Row As Integer, Col As Integer

SheetName = ActiveSheet.Name
'交互作用指定
Cells(1, 5).Value = "KougoNum ="
Cells(2, 5).Value = "KougoName ="
If Cells(1, 7).Value = "" Then
    MsgBox "KougoNum が指定されていません"
    End
End If
KougoNum = Val(Cells(1, 7).Value)
For i = 1 To KougoNum
    SpecKougo(i) = Cells(2, 6 + i).Value
    If SpecKougo(i) = "" Then
        MsgBox "KougoName が指定されていません"
        End
    End If
Next i
OrthoTitleTop = GetTitleRow("◆直交表", 1) '←Excel 列番号
'【注】直交表タイトルは"◆直交表"の行の第 3 列で設定される
rowStartXL = OrthoTitleTop + 3
colStartXL = 2 '←Excel 列番号
'直交表分岐
OrthoTabName = Cells(OrthoTitleTop, 3).Value
Select Case OrthoTabName
Case "L4+0"
    OrthoRowNum = 4 '直交表行数
    FacterNum = 3 '因子数
Case "L4+1"
    OrthoRowNum = 5 '直交表行数
    FacterNum = 3 '因子数
Case "L4+2"
    OrthoRowNum = 6 '直交表行数
    FacterNum = 3 '因子数
Case "L4+3"
    OrthoRowNum = 7 '直交表行数
    FacterNum = 3 '因子数
Case "L4+4"
    OrthoRowNum = 8 '直交表行数
    FacterNum = 3 '因子数
Case "L6(2x3)" '全組合せ実験
    AllCombExp = True
    OrthoRowNum = 6 '直交表行数
    FacterNum = 2 '因子数
Case "L8(2x4)"
    AllCombExp = True
    OrthoRowNum = 8 '直交表行数
    FacterNum = 2 '因子数
Case "L9(3x3)"
    AllCombExp = True
    OrthoRowNum = 9 '直交表行数
    FacterNum = 2 '因子数
Case "L8+0"
    OrthoRowNum = 8 '直交表行数
    FacterNum = 7 '因子数
Case "L8+1"
    OrthoRowNum = 9 '直交表行数
    FacterNum = 7 '因子数
Case "L8+2"
```

```
    OrthoRowNum = 10 '直交表行数
    FacterNum = 7 '因子数
Case "L8+3(2x4)"
    OrthoRowNum = 11 '直交表行数
    FacterNum = 5 '因子数
Case "L8+4(2x4)"
    OrthoRowNum = 12 '直交表行数
    FacterNum = 5 '因子数
Case "L8+21"
    OrthoRowNum = 29 '直交表行数
    FacterNum = 7 '因子数
Case "L9+0"
    OrthoRowNum = 9 '直交表行数
    FacterNum = 4 '因子数
Case "L9+2"
    OrthoRowNum = 11 '直交表行数
    FacterNum = 4 '因子数
Case "L9+4"
    OrthoRowNum = 13 '直交表行数
    FacterNum = 4 '因子数
Case "L9+6"
    OrthoRowNum = 15 '直交表行数
    FacterNum = 4 '因子数
Case "L18Whole" 'for display of whole Kougo columns
    OrthoRowNum = 18 '直交表行数
    FacterNum = 8 '因子数
Case "L18+0r" '追加行なしで交互作用列１個
    OrthoRowNum = 18 '直交表行数
    FacterNum = 8 '因子数
Case "L18+0s" '因子Ｈ列を削り代わりに交互作用１個を充てる
    OrthoRowNum = 18 '直交表行数
    FacterNum = 7 '因子数
Case "L18+0"
    OrthoRowNum = 18 '直交表行数
    FacterNum = 8 '因子数
Case "L18+1"
    OrthoRowNum = 19 '直交表行数
    FacterNum = 8 '因子数
Case "L18+2"
    OrthoRowNum = 20 '直交表行数
    FacterNum = 8 '因子数
Case "L18+3"
    OrthoRowNum = 21 '直交表行数
    FacterNum = 8 '因子数
Case "L18+4"
    OrthoRowNum = 22 '直交表行数
    FacterNum = 8 '因子数
Case "L18+6"
    OrthoRowNum = 24 '直交表行数
    FacterNum = 8 '因子数
Case "L18+8"
    OrthoRowNum = 26 '直交表行数
    FacterNum = 8 '因子数
Case "L18+9"
    OrthoRowNum = 27 '直交表行数
    FacterNum = 8 '因子数
Case "L18+18"
    OrthoRowNum = 36 '直交表行数
    FacterNum = 8 '因子数
Case "L18+36"
```

```
    OrthoRowNum = 54 '直交表行数
    FacterNum = 8 '因子数
Case Else
    MsgBox "Fatal error in InitialSet() No.1"
    End
End Select
If FacterNum <= 0 Or OrthoRowNum <= 0 Then
    MsgBox "InitialSet 不備です"
    End
End If
SearchCondTop = rowStartXL + OrthoRowNum + 1 '"◆探索条件"のExcel行
SetMitisuTop = rowStartXL + OrthoRowNum + 10 '"◆未知数"のExcel行
'直交表の右側をクリア
Range(Cells(OrthoTitleTop, FacterNum + KougoNum + colStartXL + 1), _
                Cells(OrthoTitleTop + OrthoRowNum + 3, 255)).Select
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
Selection.Borders(xlEdgeLeft).LineStyle = xlNone
Selection.Borders(xlEdgeTop).LineStyle = xlNone
Selection.Borders(xlEdgeBottom).LineStyle = xlNone
Selection.Borders(xlEdgeRight).LineStyle = xlNone
Selection.Borders(xlInsideVertical).LineStyle = xlNone
Selection.Borders(xlInsideHorizontal).LineStyle = xlNone
Selection.HorizontalAlignment = xlCenter
Selection.ClearContents
'直交表主効果部の読み取り
For k = 1 To OrthoRowNum
    For f = 1 To FacterNum
        OrthoTab(k, f) = _
            Val(Cells(k + rowStartXL - 1, f + colStartXL - 1).Value)
    Next f
Next k
For f = 1 To FacterNum + KougoNum
    LevelNum(f) = Cells(rowStartXL + OrthoRowNum, _
                                            colStartXL + f - 1).Value
Next f
'探索条件と探索範囲
If AllCombExp <> True Then
    '探索条件以下をクリア
    Range(Cells(SearchCondTop + 1, 4), Cells(SearchCondTop + 4, 256)).Select
    Selection.ClearContents
    Selection.HorizontalAlignment = xlCenter
    Range(Cells(SearchCondTop + 5, 1), Cells(2000, 256)).Select
    Selection.ClearContents
    Selection.HorizontalAlignment = xlCenter
    Cells(SearchCondTop, 1).Value = "◆探索条件"
    Cells(SearchCondTop + 1, 1).Value = "SearchRowStart"
    Cells(SearchCondTop + 2, 1).Value = "SearchRowEnd"
    Cells(SearchCondTop + 3, 1).Value = "SearchColStart"
    Cells(SearchCondTop + 4, 1).Value = "SearchColEnd"
    Range(Cells(SearchCondTop + 1, 1), Cells(SearchCondTop + 4, 1)).Select
    Selection.HorizontalAlignment = xlLeft
    Selection.Font.Bold = True
    '探索範囲(直交表行列番号)
    SearchRowStart = Val(Cells(SearchCondTop + 1, 3))
    SearchRowEnd = Val(Cells(SearchCondTop + 2, 3))
    SearchColStart = Val(Cells(SearchCondTop + 3, 3))
    SearchColEnd = Val(Cells(SearchCondTop + 4, 3))
    If SearchRowStart < 1 Then
        SearchRowStart = OrthoRowNum
```

```
        Cells(SearchCondTop + 1, 1).Value = "SearchRowStart"
        Cells(SearchCondTop + 1, 3).Value = SearchRowStart
    End If
    If SearchRowEnd < SearchRowStart Then
        SearchRowEnd = SearchRowStart
        Cells(SearchCondTop + 2, 1).Value = "SearchRowEnd"
        Cells(SearchCondTop + 2, 3).Value = SearchRowEnd
    End If
    If SearchColStart < 1 Then
        SearchColStart = FacterNum - 1
        Cells(SearchCondTop + 3, 1).Value = "SearchColStart"
        Cells(SearchCondTop + 3, 3).Value = SearchColStart
    End If
    If SearchColEnd < SearchColStart Then
        SearchColEnd = FacterNum
        Cells(SearchCondTop + 4, 1).Value = "SearchColEnd"
        Cells(SearchCondTop + 4, 3).Value = SearchColEnd
    End If
    If SearchColEnd < SearchColStart Then
        MsgBox "SearchColEnd 値が不適切です"
        End
    End If
    If SearchMode = True Then
        ResultTop = SetMitisuTop + 4 '"◆解リスト"の Excel 行
        ResultLeft = colStartXL
        cutColWidth = SearchColEnd - SearchColStart + 1
        cutRowWidth = SearchRowEnd - SearchRowStart + 1
        '探索範囲表示
        DispSearchArea
        'LN 進数定数セット
        c0 = "": c1 = "": c2 = "": c3 = ""
        For i = 1 To SearchRowEnd - SearchRowStart + 1
            For j = 1 To SearchColEnd - SearchColStart + 1
                c0 = c0 & "0"
                c1 = c1 & "1"
                c2 = c2 & "2"
                c3 = c3 & "3"
            Next j
        Next i
        If SearchNum <= cntUstop Then
            cntUstop = SearchNum - 1 'cntU は 0 からカウント
        End If
        Cells(SearchCondTop + 5, 1).Value = "SearchNum =" & Str(SearchNum)
        Cells(SearchCondTop + 5, 1).HorizontalAlignment = xlLeft
        'スタート&ストップ
        Cells(SearchCondTop + 6, 1).Value = "cntUstart/stop =" _
                                & Str(cntUstart) & " ～" & Str(cntUstop)
        Cells(SearchCondTop + 6, 1).HorizontalAlignment = xlLeft
    End If
Else
    '探索条件以下をクリア
    Range(Cells(SearchCondTop + 1, 1), Cells(SearchCondTop + 4, 256)).Select
    Selection.ClearContents
    Selection.HorizontalAlignment = xlCenter
    Range(Cells(SearchCondTop + 5, 1), Cells(2000, 256)).Select
    Selection.ClearContents
    Selection.HorizontalAlignment = xlCenter
    Cells(SearchCondTop, 1).Value = "◆探索条件：  " _
                & "全組合せ実験(AllCombExp = True)では探索できません"
End If
```

```
End Sub

Sub DispSearchArea()
'Display of Search area
    Dim j As Integer

    If SearchMode <> True Then Exit Sub
    '探索範囲の太線囲み
    Range(Cells(rowStartXL + SearchRowStart - 1, colStartXL + _
            SearchColStart - 1), Cells(rowStartXL + SearchRowEnd - 1, _
            colStartXL + SearchColEnd - 1)).Select
    Selection.Borders(xlDiagonalDown).LineStyle = xlNone
    Selection.Borders(xlDiagonalUp).LineStyle = xlNone
    Selection.Borders(xlEdgeLeft).LineStyle = xlNone
    Selection.Borders(xlEdgeTop).LineStyle = xlNone
    Selection.Borders(xlEdgeBottom).LineStyle = xlNone
    Selection.Borders(xlEdgeRight).LineStyle = xlNone
    Selection.Borders(xlInsideVertical).LineStyle = xlNone
    Selection.Borders(xlInsideHorizontal).LineStyle = xlNone
    Selection.Borders(xlEdgeLeft).Weight = xlMedium
    Selection.Borders(xlEdgeTop).Weight = xlMedium
    Selection.Borders(xlEdgeBottom).Weight = xlMedium
    Selection.Borders(xlEdgeRight).Weight = xlMedium
    '探索範囲に１をセット
    For k = SearchRowStart + rowStartXL - 1 To SearchRowEnd + rowStartXL - 1
        For j = SearchColStart + colStartXL - 1 To SearchColEnd + colStartXL - 1
            Cells(k, j).Value = 1
        Next j
    Next k
End Sub

Sub DispResult(cntP As Long, cntU As Long)
    Dim x As Double, j As Integer

    If cntP = 0 Then Exit Sub
    Cells(ResultTop + cntP + 1, 1).Value = cntP
    Cells(ResultTop + cntP + 1, 2).Value = cntU
    For j = 1 To cutRowWidth
        Cells(ResultTop + cntP + 1, j + 2).Value = V1(j)
    Next j
    For MitisuNr = 1 To MitisuNum
        x = kMatrix(MitisuNr, MitisuNum + 1)
        Cells(ResultTop + cntP + 1, MitisuNr + cutRowWidth + 2).Value = x
    Next MitisuNr
End Sub

Sub DispCalcCondition()
    Dim ZeroEqsNum As Integer
    Dim Margin As Integer

    OrthoEqsNum = OrthoRowNum
    ZeroEqsNum = FacterNum + KougoNum
    Margin = OrthoEqsNum + ZeroEqsNum - MitisuNum
    If Margin < 0 Then
        MsgBox "式数が未知数個数より少ないので計算を打ち切ります。"
        End
    End If
    Cells(SearchCondTop + 1, 12).Value = "OrthoEqsNum =" & Str(OrthoEqsNum)
    Cells(SearchCondTop + 1, 12).HorizontalAlignment = xlLeft
    Cells(SearchCondTop + 2, 12).Value = "ZeroEqsNum =" & Str(ZeroEqsNum)
```

```
    Cells(SearchCondTop + 2, 12).HorizontalAlignment = xlLeft
    Cells(SearchCondTop + 3, 12).Value = "MitisuNum =" & Str(MitisuNum)
    Cells(SearchCondTop + 3, 12).HorizontalAlignment = xlLeft
    Cells(SearchCondTop + 4, 12).Value = "Margin =" & Str(OrthoEqsNum) & "+" & _
                    LTrim(Str(ZeroEqsNum)) & "-" & LTrim(Str(MitisuNum)) & " =" & Str(Margin)
    Cells(SearchCondTop + 4, 12).HorizontalAlignment = xlLeft
    Cells(SearchCondTop + 1, 15).Value = "YkMode = " & YkMode
    Cells(SearchCondTop + 1, 15).HorizontalAlignment = xlLeft
    Cells(SearchCondTop + 2, 15).Value = "MitisuMode = " & MitisuMode
    Cells(SearchCondTop + 2, 15).HorizontalAlignment = xlLeft
End Sub

Sub LESQ_V2(AMN, BM, B, M1 As Integer, N1 As Integer, NSTOP As Integer)
'Least SQuare calculation
    Dim hk As Integer, j As Integer, jm As Integer
    Dim z() As Double, BN() As Double, BX() As Double
    ReDim BX(N1, N1), BN(N1), z(N1)

    For hk = 1 To N1
        For j = 1 To N1
            BX(hk, j) = 0
            For jm = 1 To M1
                BX(hk, j) = BX(hk, j) + AMN(jm, hk) * AMN(jm, j)
            Next jm
        Next j
        z(hk) = 0
        For jm = 1 To M1
            z(hk) = z(hk) + AMN(jm, hk) * BM(jm)
            BN(hk) = z(hk)
        Next jm
    Next hk
    LINSW_V2 BX, BN, N1, NSTOP
    If NSTOP = 0 Then '計算が正常終了
        For hk = 1 To N1
            BM(hk) = BN(hk)
            B(hk) = BN(hk)
        Next hk
    End If
End Sub

Public Sub LINSW_V2(A, AN, N1 As Integer, NSTOP As Integer)
'Linear Equations Sweep-out
    Dim hk As Integer, KK As Integer, M As Integer
    Dim Pvt As Double, swp As Double
    Dim B() As Integer, c() As Double
    ReDim B(N1), c(N1)

    If N1 > MaxN1 Or N1 <= 0 Then
        MsgBox ("方程式の次元数が大きすぎか不正です。")
        Exit Sub
    End If
    For hk = 1 To N1
        B(hk) = hk
    Next hk
    For hk = 1 To N1
        PivotH A, AN, B, N1, hk
        Pvt = A(hk, hk)
        If Pvt < ESC Then
            NSTOP = 9
            NRANK = hk - 1
```

```
            Exit Sub
        Else
            For M = hk To N1
                A(hk, M) = A(hk, M) / Pvt
            Next M
            AN(hk) = AN(hk) / Pvt
            For KK = 1 To N1
                If KK <> hk Then
                    swp = A(KK, hk)
                    For M = hk To N1
                        A(KK, M) = A(KK, M) - swp * A(hk, M)
                    Next M
                    AN(KK) = AN(KK) - swp * AN(hk)
                End If
            Next KK
        End If
    Next hk
    NRANK = hk - 1
    For hk = 1 To N1
        c(hk) = AN(hk)
    Next hk
    For hk = 1 To N1
        AN(B(hk)) = c(hk)
    Next hk
End Sub

Sub PivotH(A, AN, B, N1 As Integer, hk As Integer)
'Pivot calculation
    Dim i As Integer, M As Integer, BB As Integer
    Dim max_a As Double, max_i, max_j As Integer
    Dim c() As Double, ANN As Double
    ReDim c(N1)

    max_a = Abs(A(hk, hk))
    max_i = hk
    For i = hk + 1 To N1
        If Abs(A(i, hk)) > max_a Then
            max_a = Abs(A(i, hk))
            max_i = i
        End If
    Next i
    If max_i <> hk Then
        For M = 1 To N1
            c(M) = A(hk, M)
            A(hk, M) = A(max_i, M)
            A(max_i, M) = c(M)
            ANN = AN(hk)
            AN(hk) = AN(max_i)
            AN(max_i) = ANN
        Next M
    End If
    max_a = Abs(A(hk, hk))
    max_j = hk
    For i = hk + 1 To N1
        If Abs(A(hk, i)) > max_a Then
            max_a = Abs(A(hk, i))
            max_j = i
        End If
    Next i
    If max_j <> hk Then
```

```
        For M = 1 To N1
            c(M) = A(M, hk)
            A(M, hk) = A(M, max_j)
            A(M, max_j) = c(M)
            BB = B(hk)
            B(hk) = B(max_j)
            B(max_j) = BB
        Next M
    End If
End Sub

Sub Keisen(row1 As Integer, col1 As Integer, row2 As Integer, col2 As Integer)
    Range(Cells(row1, col1), Cells(row2, col2)).Select
    Selection.Borders(xlDiagonalDown).LineStyle = xlNone
    Selection.Borders(xlDiagonalUp).LineStyle = xlNone
    Selection.Borders(xlEdgeLeft).LineStyle = xlContinuous
    Selection.Borders(xlEdgeTop).LineStyle = xlContinuous
    Selection.Borders(xlEdgeBottom).LineStyle = xlContinuous
    Selection.Borders(xlEdgeRight).LineStyle = xlContinuous
    If col1 <> col2 Then
        Selection.Borders(xlInsideVertical).LineStyle = xlContinuous
    End If
    Selection.Borders(xlInsideHorizontal).LineStyle = xlContinuous
End Sub

Sub MakeEqs()
'List up of the simultanious linear equations
'連立方程式の式をリストアップします。係数行列表の変数名を用います。
    Dim cnt As Integer, i As Integer, j As Integer, ii As Integer
    Dim Eqs As String

    EqsTop = KeisuMatrixTop + OrthoRowNum + FacterNum + KougoNum
    Cells(EqsTop, 1).Value = "◆連立方程式（実験値式&ゼロ和式）" _
                                    & "  式は係数行列に基づいて作成"
    Cells(EqsTop, 1).HorizontalAlignment = xlLeft
    Cells(EqsTop, 1).Select
    M1 = OrthoRowNum + FacterNum + KougoNum
    N1 = M1
    For k = 1 To M1
        Eqs = "y(" & Cells(KeisuMatrixTop + k - 1, 1).Value & ") = "
        cnt = 0
        For j = 1 To N1
            If kMatrix(k, j) = 1 Then
                cnt = cnt + 1
                If cnt = 1 Then
                    Eqs = Eqs & Cells(KeisuMatrixTop - 1, j + 1).Value
                Else
                    Eqs = Eqs & " + " & Cells(KeisuMatrixTop - 1, j + 1).Value
                End If
            ElseIf kMatrix(k, j) = -1 Then
                cnt = cnt + 1
                If cnt = 1 Then
                    Eqs = Eqs & Cells(KeisuMatrixTop - 1, j + 1).Value
                Else
                    Eqs = Eqs & " - " & Cells(KeisuMatrixTop - 1, j + 1).Value
                End If
            End If
        Next j
        Eqs = Eqs & " = " & Int(Val(y(k) * 1000)) / 1000
        Cells(EqsTop + k, 2).Value = Eqs
```

```
        Cells(EqsTop + k, 2).HorizontalAlignment = xlLeft
    Next k
End Sub

Function GetTitleRow(TitleString As String, TitleCol As Integer)
'タイトル文字列をTitleCol列で探索。見つかれば行番号を返します。
    Dim i As Integer

    For i = 1 To 1000
        If Cells(i, TitleCol).Value = TitleString Then
            GetTitleRow = i
            Exit Function
        End If
    Next i
    MsgBox TitleString & "が見つかりません"
    End
End Function

Sub DispOrthoTab()
'Display the ortogonal table
    Dim k As Integer, j As Integer

    For k = 1 To OrthoRowNum
        For j = 1 To FacterNum + KougoNum
            Cells(rowStartXL + k - 1, colStartXL + j - 1).Value = OrthoTab(k, j)
        Next j
    Next k
End Sub

Function Hantei() As Boolean
'連立方程式式で解あり／なしを判定
    If NSTOP = 0 Then
        If YkMode <> "HAND" Then
            For MitisuNr = 1 To MitisuNum
                If Abs(MitisuVal(MitisuNr) - _
                                kMatrix(MitisuNr, MitisuNum + 1)) > ESC Then
                    Hantei = False
                    Exit Function
                End If
            Next MitisuNr
        End If
        Hantei = True
    Else
        Hantei = False
        Exit Function
    End If
End Function
```

６．プログラムソース

```
Option Explicit
'==========================================================================
'Search32.09.xls (選択分岐は①探索、②解析の２ケースだけに変更)
'    Search of Expanded Orthogonal Table (拡張直交表探索)
'2011.05.13.Y.Tanaka, K.Horino
'==========================================================================
Const ESC = 0.0000000001
Const MaxN1 = 100 'max number of mitisu
Const KDF = 100# 'KougoDefaultFacter using as MitisuVal(mNr) = mNr / KDF
Const ScreenUpdate = False

'Q-break (push Q-key, you can break)←use Windows API
Private Declare Function GetAsyncKeyState _
                      Lib "User32.dll" (ByVal vKey As Long) As Long
'InitialSet
Dim SheetName As String, OrthoRowNum As Integer '
Dim rowNum As Integer, FacterNum As Integer, KougoNum As Integer
Dim rowStartXL As Double, colStartXL As Double
Dim SearchCondTop As Integer, OrthoTitleTop As Integer
'sweep out
Dim ResultLeft As Integer, ResultTop As Integer
Dim N As Integer, NRANK As Integer, NSTOP As Integer
'OrthoTab
Dim OrthoTab(MaxN1, MaxN1) As Double, LevelNum(MaxN1) As Integer
Dim OrthoTabName As String
Dim f As Integer, ff As Integer, k As Integer
Dim cntUstart As Long, cntUstop As Long, cntUbreak As Long
'MitisuSet
Dim MitisuMode As String, MitisuNr As Integer, MitisuNum As Integer
Dim MitisuVal(MaxN1) As Double, MitisuName(MaxN1) As String
Dim OrthoFacterName(MaxN1) As String, OrthoKougoName(MaxN1) As String
Dim SetMitisuTop As Integer, KougoNr As Integer
Dim M1 As Integer, N1 As Integer
'KeisuMatrix
Dim KeisuMatrixTop As Integer, KeisuMatrixLeft As Integer
Dim kMatrix(MaxN1, MaxN1) As Double
'ykCalc
Dim YkMode As String, y(MaxN1) As Double
'search
Dim SearchRowStart As Integer, SearchRowEnd As Integer
Dim SearchColStart As Integer, SearchColEnd As Integer
Dim LN As Integer
Dim c0 As String, c1 As String, c2 As String, c3 As String
Dim U As String, V As String
Dim vv As String
Dim SearchMode As Boolean
Dim cutColWidth As Integer, cutRowWidth As Integer
Dim V1(MaxN1) As String
Dim cntP As Long, cntU As Long 'パス（合格）カウント，サーチカウント
Dim SearchNum As Long
'specification of Kougo (解析交互作用の指定)
Dim SpecKougo(MaxN1) As String, SearchAreaLevelNum(MaxN1) As Integer
'MakeEqs (make the linear simultanious equations)
Dim EqsTop As Integer
Dim FacterName(20) As String
'ZeroSum(ゼロ和法)&AllCombExp (全組実験)
Dim AllCombExp As Boolean 'set at InitialSet() 全組合せ実験では探索しない
Dim ZeroEqsNum As Integer, OrthoEqsNum As Integer
'skip of sub YkCalc
```

```
Dim skipYkCalc As Boolean

Sub Main()
    Dim fNr As Integer, lineNr As Integer
    Dim i As Integer, ii As Integer, j As Integer
    Dim Maebu(61) As String, Atobu(61) As String
    Dim cEnd As String, msg As String, Row As Integer, Col As Integer
    Dim HanteiResult As Boolean, yesno As Integer, aa As String
    Dim B(MaxN1) As Double, BM(MaxN1) As Double
    Dim now0 As Variant, now1 As Variant
    Dim timer0 As Variant, timer1 As Variant, TimerSec As Long

'*****************************************************************************
MsgBox "ActiveSheet.Name = " & ActiveSheet.Name
'*****************************************************************************

    Application.StatusBar = ""
    cntU = 0 '【注】ないと入力未知数欄が表示されません
    cntP = 0
    'チェック(直交表仕上げまで先行)
    SearchMode = False '前回計算の SearchMode 値を消去
    InitialSet cntUstart, cntUstop
    OrthoTabKougo
    '探索／解析計算の選択
    yesno = MsgBox("SearchMode ?" & Chr(10) & " はい⇒探索計算" _
                        & Chr(10) & " いいえ⇒解析計算" _
                        & Chr(10) & " ｷｬﾝｾﾙ ⇒中止", vbYesNoCancel)
    Select Case yesno
    Case vbCancel
    Case vbNo
        Cells(1, 1).Value = "解析計算"
        YkMode = "HAND"
        'y(k)読み込みチェック
        For k = 1 To OrthoRowNum
            If Cells(k + rowStartXL - 1, _
                    FacterNum + KougoNum + 2).Value = "" Then
                MsgBox "直交表右端列の y(k)値が記入されてません"
                End
            End If
        Next k
        MitisuMode = "HAND"
        For k = 1 To OrthoRowNum
            y(k) = Val(Cells(k + rowStartXL - 1, _
                        FacterNum + KougoNum + 2).Value)
        Next k
    Case vbYes
        Cells(1, 1).Value = "探索計算"
        YkMode = "AUTO"
        skipYkCalc = True
        MitisuMode = "AUTO"
    Case Else
        MsgBox "error in Main() No.4"
        End
    End Select
    Select Case yesno
    Case vbCancel
        End
    Case vbNo
        '単発計算
        Erase OrthoTab
```

```
    SearchMode = False
    cntU = 0 'ないと入力未知数欄が表示されません
    InitialSet cntUstart, cntUstop
    OrthoTabKougo
    MitisuValSet
    SetMitisu
    KeisuMatrix
    YkCalc
    GoSub LESQcalc
    If Hantei = True Then
        msg = "合格 NRANK =" & Str(NRANK) & " NSTOP =" & Str(NSTOP)
    Else
        msg = "不合格 NRANK =" & Str(NRANK) & " NSTOP =" & Str(NSTOP)
    End If
    Cells(SetMitisuTop + 4, 3).Value = msg
    Cells(SetMitisuTop + 4, 3).HorizontalAlignment = xlLeft
    Cells(SetMitisuTop + 4, 3).Select
    MakeEqs
    MsgBox "計算終了  " & msg
    End
Case vbYes
    SearchMode = True
    '探索計算開始
    SearchNum = 1
    For Row = SearchRowStart To SearchRowEnd
        For Col = SearchColStart To SearchColEnd
            SearchNum = SearchNum * LevelNum(Col)
        Next Col
    Next Row
    Cells(SearchCondTop + 5, 1).Value = "SearchNum =" & Str(SearchNum)
    Cells(SearchCondTop + 5, 1).HorizontalAlignment = xlLeft
    Cells(SearchCondTop + 6, 3).Select
    cntUstart = Val(InputBox("スタート・ランNo. cntU = ?", , "0"))
    cntUstop = Val(InputBox("エンド・ランNo. cntUstop = ?", , "0"))
    If SearchNum < cntUstop Then
        cntUstop = SearchNum
    End If
    InitialSet cntUstart, cntUstop
    GoSub time1
    '最初は直交表の表示を仕上げます
    OrthoTabKougo
    DispSearchArea
    MitisuValSet
    YkCalc
    SetMitisu
    KeisuMatrix
    '画面更新の停止
    Application.ScreenUpdating = ScreenUpdate
    '解リスト（モニター）項目行
    cntU = 0
    If cntU = 0 Then
        If SearchMode = True Then
            Cells(ResultTop, 1).Value = "◆解リスト"
            Cells(ResultTop, 1).HorizontalAlignment = xlLeft
            Cells(ResultTop + 1, 1).Value = "cntP"
            Cells(ResultTop + 1, 2).Value = "cntU"
            For k = SearchRowStart To SearchRowEnd
                Cells(ResultTop + 1, k - SearchRowStart + 3).Value _
                                                = "k =" & Str(k)
            Next k
```

```
        For MitisuNr = 1 To MitisuNum
            Cells(ResultTop + 1, MitisuNr + cutRowWidth + 2).Value _
                                        = MitisuName(MitisuNr)
        Next MitisuNr
    End If
End If
'網羅的探索
cntU = cntUstart
cntP = 0
If cntU < 0 Then
    cntU = 0
End If
U = Change10toLN(cntU, LN)
If SearchNum < cntUstop Then
    cntUstop = SearchNum
End If
Do
    U = Change10toLN(cntU, LN)
    vv = ChangeUtoV(U) 'vv は探索領域シリアル
    For lineNr = SearchRowStart To SearchRowEnd
        ii = lineNr - SearchRowStart + 1
        i = cutColWidth * (ii - 1) + 1
        'vv を探索部 1 行ずつに分けます
        V1(ii) = Mid(vv, i, cutColWidth)
        'OrthoTab 更新
        For fNr = SearchColStart To SearchColEnd
            OrthoTab(lineNr, fNr) = Val(Mid(V1(ii), _
                    fNr - SearchColStart + 1, 1))
        Next fNr
    Next lineNr
    OrthoTabKougo
    DispOrthoTab
    YkCalc
    KeisuMatrix
    GoSub LESQcalc
    '解の判定
    If Hantei = True And cntU >= 0 Then
        cntP = cntP + 1
        DispResult cntP, cntU
    End If
    Application.StatusBar = "cntU =" & Str(cntU) & _
                        ", cntP =" & Str(cntP) & ", vv = " & vv
    cntU = cntU + 1 '次回の準備
    '"Q"キー(Quit)強制中断の処置
    If GetAsyncKeyState(vbKeyQ) Then
        Application.ScreenUpdating = True
        U = Change10toLN(cntU, LN)
        vv = ChangeUtoV(U) 'vv は探索領域シリアル
        GoSub BreakCalc
        If MsgBox("OK:終了, ｷｬﾝｾﾙ:継続", vbOKCancel) = vbOK Then
            Exit Do
        End If
        Application.ScreenUpdating = ScreenUpdate
    End If
Loop Until cntU > cntUstop
cntU = cntU - 1
Cells(SearchCondTop + 6, 1).Value = "cntUstart/stop =" _
                            & Str(cntUstart) & " ～" & Str(cntU)
Cells(SearchCondTop + 6, 1).HorizontalAlignment = xlLeft
U = Change10toLN(cntU, LN)
```

```
        vv = ChangeUtoV(U) 'vvは探索領域シリアル
        GoSub BreakCalc
        If SearchMode = True Then
            GoSub time2
        End If
        '画面更新の復活
        Application.ScreenUpdating = True
        MsgBox "探索計算終了"
        Application.StatusBar = ""
    Case Else
        MsgBox "error in Main() No.4"
        End
    End Select
    End
Exit Sub

BreakCalc:
    '最終y(k)と直交表を記入（探索領域行のみ）
    OrthoTabKougo
    DispOrthoTab
    YkCalc
    If skipYkCalc = False Then
        For lineNr = SearchRowStart To SearchRowEnd
            Cells(lineNr + rowStartXL - 1, _
                    FacterNum + KougoNum + 2).Value = y(lineNr)
        Next lineNr
    End If
    cntUbreak = cntU
    Cells(SearchCondTop + 5, 6).Value = "cntP =" & Str(cntP)
    Cells(SearchCondTop + 5, 6).HorizontalAlignment = xlLeft
    Cells(SearchCondTop + 6, 6).Value = "Hit% =" & Str(Int(cntP _
            / (cntUbreak - cntUstart + 1) * 100# * 100#)) / 100# & "%"
    Cells(SearchCondTop + 6, 6).HorizontalAlignment = xlLeft
    GoSub time2
Return

time1:
    now0 = Now
    timer0 = Timer
    Cells(SearchCondTop + 1, 6).Value = "START"
    Cells(SearchCondTop + 1, 6).HorizontalAlignment = xlLeft
    Cells(SearchCondTop + 1, 7).Value = " " & now0
    Cells(SearchCondTop + 1, 7).HorizontalAlignment = xlLeft
Return

time2:
    now1 = Now
    timer1 = Timer
    TimerSec = (timer1 - timer0) + Int(now1 - now0) * 86400
    Cells(SearchCondTop + 2, 6).Value = "END"
    Cells(SearchCondTop + 2, 6).HorizontalAlignment = xlLeft
    Cells(SearchCondTop + 2, 7).Value = " " & Now
    Cells(SearchCondTop + 2, 7).HorizontalAlignment = xlLeft
    '所用時間
    TimerSec = (timer1 - timer0) + Int(now1 - now0) * 86400
    Cells(SearchCondTop + 3, 6).Value = "TimerSec =" & Str(TimerSec)
    Cells(SearchCondTop + 3, 6).HorizontalAlignment = xlLeft
    Cells(SearchCondTop + 4, 6).Value = "Velocity"
    Cells(SearchCondTop + 4, 7).Value = Str(1000 * CDbl(TimerSec) _
                        / CDbl(cntU - cntUstart + 1))
```

```
    Cells(SearchCondTop + 4, 7).HorizontalAlignment = xlCenter
    Cells(SearchCondTop + 4, 8).NumberFormatLocal = "0.0_ "
    Cells(SearchCondTop + 4, 8).Value = "msec／run"
    Cells(SearchCondTop + 4, 8).HorizontalAlignment = xlLeft
Return

LESQcalc:
    For k = 1 To rowNum
        BM(k) = y(k)
    Next k
    NSTOP = 0
    LESQ_V2 kMatrix, BM, B, rowNum, MitisuNum, NSTOP
    For MitisuNr = 1 To MitisuNum
        kMatrix(MitisuNr, MitisuNum + 1) = B(MitisuNr)
        Cells(SetMitisuTop + 3, MitisuNr + 1).Value = B(MitisuNr)
        Cells(SetMitisuTop + 3, MitisuNr + 1).HorizontalAlignment = xlCenter
    Next MitisuNr
Return
End Sub

Sub SetMitisu()
'未知数設定
    Dim j As Integer

    Cells(SetMitisuTop, 1).Value = "◆未知数（効果基準, 因子効果,他）"
    Cells(SetMitisuTop, 1).HorizontalAlignment = xlLeft
    For j = 1 To MitisuNum
        Cells(SetMitisuTop + 1, j + 1).Value = MitisuName(j)
        If YkMode = "AUTO" Then
            Cells(SetMitisuTop + 2, j + 1).Value = MitisuVal(j)
        Else
            Cells(SetMitisuTop + 2, j + 1).Value = "･･･"
        End If
    Next j

    If MitisuMode = "AUTO" Then
        Cells(SetMitisuTop + 2, 1).Value = "AutoVal(仮設定)"
    Else
        Cells(SetMitisuTop + 2, 1).Value = "HandVal"
    End If
    Cells(SetMitisuTop + 2, 1).HorizontalAlignment = xlLeft
    Cells(SetMitisuTop + 3, 1).Value = "CalcVal(確認)"
    Cells(SetMitisuTop + 3, 1).HorizontalAlignment = xlLeft

End Sub

Sub MitisuValSet()
    Dim Level As Integer, s As Integer, j As Integer, sum As Double
    Dim EqsNum As Integer, Margin As Integer
    Dim mNr As Integer 'MitisuNr の置き換え

    mNr = 1
    MitisuName(mNr) = "μ" '←μは mNr = 1
    For f = 1 To FacterNum
        For Level = 1 To LevelNum(f)
            mNr = mNr + 1
            MitisuName(mNr) = OrthoFacterName(f) & LTrim(Str(Level))
        Next Level
    Next f
    If KougoNum > 0 Then
```

```
    For KougoNr = 1 To KougoNum
        For Level = 1 To LevelNum(FacterNum + KougoNr)
            mNr = mNr + 1
            MitisuName(mNr) = OrthoKougoName(KougoNr) _
                                        & LTrim(Str(Level))
        Next Level
    Next KougoNr
End If
MitisuNum = mNr
EqsNum = OrthoRowNum
ZeroEqsNum = FacterNum + KougoNum
Cells(SearchCondTop + 1, 11).Value = "EqsNum =" & Str(EqsNum)
Cells(SearchCondTop + 1, 11).HorizontalAlignment = xlLeft
Cells(SearchCondTop + 2, 11).Value = "ZeroEqsNum =" & Str(ZeroEqsNum)
Cells(SearchCondTop + 2, 11).HorizontalAlignment = xlLeft
Cells(SearchCondTop + 3, 11).Value = "MitisuNum =" & Str(MitisuNum)
Cells(SearchCondTop + 3, 11).HorizontalAlignment = xlLeft
Margin = EqsNum + ZeroEqsNum - MitisuNum
Cells(SearchCondTop + 4, 11).Value = "Margin =" & Str(EqsNum) & "+" & _
    LTrim(Str(ZeroEqsNum)) & "-" & LTrim(Str(MitisuNum)) _
    & " =" & Str(Margin)
Cells(SearchCondTop + 4, 11).HorizontalAlignment = xlLeft
Cells(SearchCondTop + 5, 11).Value = "AllCombExp = " & AllCombExp
Cells(SearchCondTop + 5, 11).HorizontalAlignment = xlLeft
If Margin < 0 Then
    MsgBox "式数が未知数個数より少ないので計算を打ち切ります。"
    End
End If
Cells(SearchCondTop + 6, 11).Value = ThisWorkbook.Name
Cells(SearchCondTop + 6, 11).HorizontalAlignment = xlLeft
'係数行列サイズ
M1 = rowNum
N1 = MitisuNum
If YkMode = "HAND" Then 'calculated without relation to mitisu value
    Exit Sub
End If
'未知数 HAND 設定
If MitisuMode = "HAND" Then
    Cells(3, 5).Value = "MitisuName ="
    Cells(3, 5).Font.Bold = True
    Cells(3, 5).HorizontalAlignment = xlLeft
    Cells(4, 5).Value = "HandSetVal ="
    Cells(4, 5).Font.Bold = True
    Cells(4, 5).HorizontalAlignment = xlLeft
    For MitisuNr = 1 To MitisuNum
        Cells(3, 6 + MitisuNr).Value = MitisuName(MitisuNr)
        Cells(3, 6 + MitisuNr).HorizontalAlignment = xlCenter
        If Len(Cells(4, 6 + MitisuNr).Value) = 0 Then
            Cells(4, 6 + MitisuNr).Value = 0
        End If
        MitisuVal(MitisuNr) = Val(Cells(4, 6 + MitisuNr).Value)
    Next MitisuNr
    '未知数未入力チェック
    For MitisuNr = 2 To MitisuNum
        sum = sum + MitisuVal(MitisuNr) ^ 2
    Next MitisuNr
    If sum < ESC Then
        MsgBox "第 4 行で設定未知数(HandSetVal)の値が記入されていません"
        End
    End If
```

```
ElseIf MitisuMode = "AUTO" Then
    '主効果値の自動設定
    MitisuVal(1) = 10
    mNr = 1
    For f = 1 To FacterNum
        For Level = 1 To LevelNum(f)
            mNr = mNr + 1
            Select Case LN
            Case 0
                Select Case LevelNum(f)
                Case 2
                    Select Case Level
                    Case 1
                        MitisuVal(mNr) = f
                    Case 2
                        MitisuVal(mNr) = -MitisuVal(mNr - 1)
                    Case Else
                        MsgBox "Fatal error in MitisuValSet() No.1"
                    End Select
                Case 3
                    Select Case Level
                    Case 1
                        MitisuVal(mNr) = f
                    Case 2
                        MitisuVal(mNr) = MitisuVal(mNr - 1) + 0.5
                    Case 3
                        MitisuVal(mNr) = -MitisuVal(mNr - 2) - MitisuVal(mNr - 1)
                    Case Else
                    End Select
                Case 4
                    Select Case Level
                    Case 1
                        MitisuVal(mNr) = f
                    Case 2
                        MitisuVal(mNr) = MitisuVal(mNr - 1) / 2#
                    Case 3
                        MitisuVal(mNr) = MitisuVal(mNr - 1) / 4#
                    Case 4
                        MitisuVal(mNr) = -MitisuVal(mNr - 1) _
                                    - MitisuVal(mNr - 2) - MitisuVal(mNr - 3)
                    Case Else
                        MsgBox "Fatal error in MitisuValSet() No.2"
                    End Select
                Case Else
                    MsgBox "Fatal error in MitisuValSet() No.3"
                End Select
            Case 2
                Select Case Level
                Case 1
                    MitisuVal(mNr) = f
                Case 2
                    MitisuVal(mNr) = -MitisuVal(mNr - 1)
                Case Else
                    MsgBox "Fatal error in MitisuValSet() No.4"
                End Select
            Case 3
                Select Case Level
                Case 1
                    MitisuVal(mNr) = f
                Case 2
```

```
                MitisuVal(mNr) = MitisuVal(mNr - 1) + 0.5
            Case 3
                MitisuVal(mNr) = -MitisuVal(mNr - 2) - MitisuVal(mNr - 1)
            Case Else
            End Select
        Case Else
            MsgBox "Fatal error in MitisuValSet() No.5"
        End Select
    Next Level
Next f
'交互作用値の自動設定
If KougoNum > 0 Then
    For KougoNr = 1 To KougoNum
        For Level = 1 To LevelNum(FacterNum + KougoNr)
            mNr = mNr + 1
            Select Case LN
            Case 0
                Select Case LevelNum(FacterNum + KougoNr)
                Case 2
                    Select Case Level
                    Case 1
                       MitisuVal(mNr) = CSng(mNr) / KDF
                    Case 2
                        MitisuVal(mNr) = -MitisuVal(mNr - 1)
                    Case Else
                        MsgBox "Fatal error in MitisuValSet() No.6"
                    End Select
                Case 3
                    Select Case Level
                    Case 1
                        MitisuVal(mNr) = CSng(mNr) / KDF
                    Case 2
                        MitisuVal(mNr) = MitisuVal(mNr - 1) + 0.0005
                    Case 3
                        MitisuVal(mNr) = -MitisuVal(mNr - 2) - MitisuVal(mNr - 1)
                    Case Else
                    End Select
                Case 4
                    Select Case Level
                    Case 1
                        MitisuVal(mNr) = CSng(mNr) / KDF
                    Case 2
                        MitisuVal(mNr) = MitisuVal(mNr - 1) / 2#
                    Case 3
                        MitisuVal(mNr) = MitisuVal(mNr - 1) / 4#
                    Case 4
                        MitisuVal(mNr) = -MitisuVal(mNr - 1) _
                            - MitisuVal(mNr - 2) - MitisuVal(mNr - 3)
                    Case Else
                        MsgBox "Fatal error in MitisuValSet() No.7"
                    End Select
                Case Else
                    MsgBox "Fatal error in MitisuValSet() No.8"
                End Select
            Case 2
                Select Case Level
                Case 1
                    MitisuVal(mNr) = mNr / KDF
                Case 2
                    MitisuVal(mNr) = -MitisuVal(mNr - 1)
```

```
                    Case Else
                        MsgBox "Fatal error in MitisuValSet() No.9"
                    End Select
                Case 3
                    Select Case Level
                    Case 1
                        MitisuVal(mNr) = mNr / KDF
                    Case 2
                        MitisuVal(mNr) = _
                                MitisuVal(mNr - 1) + 0.1 / 100#
                    Case 3
                    MitisuVal(mNr) = -MitisuVal(mNr - 2) - MitisuVal(mNr - 1)
                    Case Else
                    End Select
                Case 4
                    Select Case Level
                    Case 1
                        MitisuVal(mNr) = mNr / KDF
                    Case 2
                        MitisuVal(mNr) = mNr / KDF * (1# + 0.5)
                    Case 3
                        MitisuVal(mNr) = -MitisuVal(mNr - 1)
                    Case 4
                        MitisuVal(mNr) = -MitisuVal(mNr - 1) _
                                - MitisuVal(mNr - 2) - MitisuVal(mNr - 3)
                    Case Else
                        MsgBox "Fatal error in MitisuValSet() No.10"
                    End Select
                Case Else
                    MsgBox "Fatal error in MitisuValSet() No.11"
                End Select
            Next Level
        Next KougoNr
    End If
Else
    MsgBox "Error in MitisuValSet() No.12"
End If
For mNr = 2 To MitisuNum
    sum = sum + MitisuVal(mNr) ^ 2
Next mNr
If sum = 0 Then
    MsgBox "設定未知数の値が記入されていません"
    End
End If
End Sub

Sub YkCalc()
'y(k)デフォルト値の算出
    Dim Row As Integer, x As Double
    Dim j As Integer, jStart(100) As Integer

    If YkMode = "HAND" Then
        Exit Sub
    End If
    '未知数のスタート位置 MitisuNr
    jStart(1) = 2
    For j = 2 To FacterNum + KougoNum
        jStart(j) = jStart(j - 1) + LevelNum(j - 1)
    Next j
    'y(k)を計算します
```

```
        For k = 1 To OrthoRowNum
            y(k) = MitisuVal(1)
            For j = 1 To FacterNum + KougoNum
                Select Case OrthoTab(k, j)
                Case 1: x = MitisuVal(jStart(j))
                Case 2: x = MitisuVal(jStart(j) + 1)
                Case 3: x = MitisuVal(jStart(j) + 2)
                Case 4: x = MitisuVal(jStart(j) + 3)
                Case -1: x = -MitisuVal(jStart(j))
                Case -2: x = -MitisuVal(jStart(j) + 1)
                Case -3: x = -MitisuVal(jStart(j) + 2)
                Case -4: x = -MitisuVal(jStart(j) + 3)
                Case Else
                    MsgBox "Fatal error in YkCalc() No.1"
                End Select
                y(k) = y(k) + x
            Next j
            If skipYkCalc = False Then
                '直交表右端列に記入
                Cells(k + rowStartXL - 1, FacterNum + KougoNum + 2).Value = y(k)
            End If
        Next k
    For k = OrthoRowNum + 1 To M1
        y(k) = 0#
    Next k
    For k = 1 To M1
        '係数行列右端列に記入
        kMatrix(k, M1 + 1) = y(k)
    Next k
End Sub

Sub KeisuMatrix()
    Dim KeisuMatrixYoko(MaxN1) As String
    Dim j As Integer, jj As Integer, jjj As Integer, jjNum As Integer
    Dim KougoCnt As Integer, L As Integer, cnt As Integer

    Erase kMatrix
    KeisuMatrixLeft = colStartXL
    KeisuMatrixTop = SetMitisuTop + 10
    '最初全要素を0にする
    For k = 1 To M1
        For j = 1 To N1
            kMatrix(k, j) = 0
        Next j
    Next k
    '係数行列記入(1)：μ列
    For k = 1 To OrthoRowNum
        kMatrix(k, 1) = 1
    Next k
    '係数行列記入(2)：主効果
    For k = 1 To OrthoRowNum
        jj = 0
        For j = 1 To FacterNum
            jj = jj + LevelNum(j - 1)
            jjj = jj + OrthoTab(k, j) + 2 - 1
            kMatrix(k, jjj) = 1
        Next j
    Next k
    jjNum = jj '主効果の未知数個数
    '係数行列記入(3)：U&V交互作用列
```

```
    For k = 1 To OrthoRowNum
        jj = jjNum
        For j = FacterNum + 1 To FacterNum + KougoNum
            jj = jj + LevelNum(j - 1)
            jjj = jj + Abs(OrthoTab(k, j)) + 2 - 1
            If OrthoTab(k, j) > 0 Then
                kMatrix(k, jjj) = 1
            ElseIf OrthoTab(k, j) < 0 Then
                kMatrix(k, jjj) = -1
            End If
        Next j
    Next k
    '係数行列記入(4)：ゼロ和条件式
    j = 0
    For k = OrthoRowNum + 1 To OrthoRowNum + FacterNum + KougoNum
        cnt = k - OrthoRowNum
        j = j + LevelNum(cnt)
        For L = 1 To LevelNum(cnt)
            Select Case LevelNum(cnt)
            Case 2
                kMatrix(k, j + L - 1) = 1
            Case 3
                kMatrix(k, j + L - 2) = 1
            Case 4
                kMatrix(k, j + L - 3) = 1
            Case Else
                MsgBox "Fatal error in KeisuMatrix() No.1"
            End Select
        Next L
    Next k
    rowNum = OrthoRowNum + cnt
    If SearchMode <> True Then
        DispKeisuMatrix
    End If
End Sub

Sub DispKeisuMatrix()
    Dim j As Integer, jj As Integer, k As Integer

    '係数行列タテヨコ項目記入
    Cells(KeisuMatrixTop - 2, 1).Value = "◆係数行列 kMatrix(i,j)"
    Cells(KeisuMatrixTop - 2, 1).HorizontalAlignment = xlLeft
    For j = 1 To MitisuNum 'M1
        Cells(KeisuMatrixTop - 1, j + KeisuMatrixLeft - 1).Value _
                                                = MitisuName(j)
    Next j
    For k = 1 To rowNum 'M1
        Cells(k + KeisuMatrixTop - 1, 1).Value = k
    Next k
    Cells(KeisuMatrixTop - 2, 5).Value = "M1 =" & Str(MitisuNum) & _
                                    "行, N1=" & Str(rowNum) & "列"
    Cells(KeisuMatrixTop - 2, 5).HorizontalAlignment = xlLeft
    Range(Cells(KeisuMatrixTop - 1, KeisuMatrixLeft - 1), _
                                    Cells(KeisuMatrixTop + M1 - 1, _
                                    KeisuMatrixLeft + N1 - 1)).Select
    Selection.HorizontalAlignment = xlCenter
    '全要素を表示(右端 y(k)は別箇所 YkCalc で表示)
    For k = 1 To rowNum 'M1
        For j = 1 To N1
            Cells(k + KeisuMatrixTop - 1, _
```

```
                            KeisuMatrixLeft + j - 1).Value = kMatrix(k, j)
        Next j
    Next k
    Cells(KeisuMatrixTop + MitisuNum, 1).Select
End Sub

Sub OrthoFacterAndKougoName()
    Dim Level As Integer, KougoNr As Integer, cntF As Integer

    '主効果名
    cntF = 0
    For f = 1 To FacterNum
        OrthoFacterName(f) = Chr(96 + f)
        Cells(rowStartXL - 2, colStartXL + f - 1).Value _
                                        = OrthoFacterName(f)
        cntF = cntF + 1
        Cells(rowStartXL - 1, colStartXL + cntF - 1).Value = cntF
    Next f
    '交互作用名
    For KougoNr = 1 To KougoNum
        OrthoKougoName(KougoNr) = SpecKougo(KougoNr)
        Cells(rowStartXL - 2, colStartXL + FacterNum + KougoNr - 1).Value _
                                            = OrthoKougoName(KougoNr)
        Cells(rowStartXL - 1, colStartXL + cntF - 1).Value = cntF
    Next KougoNr
End Sub

Sub OrthoTabKougo()
'交互作用部作成(Kougo: interaction of orthogonal table)
    Dim kougo(5, 5) As Integer, kougoU(5, 5) As Integer
    Dim ss As Integer, cntF As Integer
    Dim c As String, LargerLevel As String, SmallerLevel As String
    Dim g As Integer, j As Integer, Col As Integer
    Dim row1 As Integer, col1 As Integer, row2 As Integer, col2 As Integer

    If cntU = 0 Then
        '未知数名記入
        OrthoFacterAndKougoName
    End If
    '交互作用部作成
    For k = 1 To OrthoRowNum
        For KougoNr = 1 To KougoNum
            f = Asc(Mid(SpecKougo(KougoNr), 1, 1)) - 96
            ff = Asc(Mid(SpecKougo(KougoNr), 2, 1)) - 96
            If f < 0 Or ff < 0 Then
                MsgBox "交互作用名 " & SpecKougo(KougoNr) _
                                        & " を半角にしてください"
                End
            End If
            If LevelNum(f) > LevelNum(ff) Then
                LargerLevel = LTrim(Str(LevelNum(f)))
                SmallerLevel = LTrim(Str(LevelNum(ff)))
            Else
                LargerLevel = LTrim(Str(LevelNum(ff)))
                SmallerLevel = LTrim(Str(LevelNum(f)))
            End If
            c = SmallerLevel & "×" & LargerLevel
            GoSub kougoDef
            '交互作用部記入
            j = FacterNum + KougoNr
```

```
            OrthoTab(k, j) = kougo(OrthoTab(k, f), OrthoTab(k, ff))
            LevelNum(j) = Val(LargerLevel)
            If cntU = 0 Then
                Cells(k + rowStartXL - 1, KougoNr + FacterNum + 1).Value = _
                                    kougo(OrthoTab(k, f), OrthoTab(k, ff))
                Cells(OrthoRowNum + rowStartXL, KougoNr + _
                                    FacterNum + 1).Value = LevelNum(j)
            End If
            If cntU = 0 Then
                '交互作用部のｆ番号
                Cells(rowStartXL - 1, FacterNum + KougoNr + 1).Value _
                                          = FacterNum + KougoNr
            End If
        Next KougoNr
    Next k
    '直交表右端へ"y(k)"記入
    Cells(rowStartXL - 2, colStartXL + FacterNum _
                                    + KougoNum).Value = "y(k)"
    If cntU = cntUstart Then
        '交互作用部罫線
        If KougoNum > 0 Then
            row1 = rowStartXL
            row2 = row1 + OrthoRowNum - 1
            col1 = colStartXL + FacterNum
            col2 = col1 + KougoNum - 1
            Keisen row1, col1, row2, col2
        End If
    End If
    Cells(SearchCondTop + 5, 1).Select
Exit Sub

kougoDef: '交互作用定義
    Select Case c
    Case "2×2"
        Select Case Right(SpecKougo(KougoNr), 1)
        Case "U", "V"
            MsgBox "交互作用名末尾に不要のＵまたはＶがあります"
            End
        Case Else
            kougo(1, 1) = 1
            kougo(1, 2) = 2
            kougo(2, 1) = 2
            kougo(2, 2) = 1
        End Select
    Case "2×3"
        If Right(SpecKougo(KougoNr), 1) <> "W" Then
            MsgBox "KougoName の末尾""W""をつけなさい"
            End
        End If
        kougo(1, 1) = 1
        kougo(1, 2) = 2
        kougo(1, 3) = 3
        kougo(2, 1) = -1
        kougo(2, 2) = -2
        kougo(2, 3) = -3
    Case "2×4"
        If Right(SpecKougo(KougoNr), 1) <> "Z" Then
            MsgBox "KougoName の末尾""Z""をつけなさい"
            End
        End If
```

```
        kougo(1, 1) = 1
        kougo(2, 1) = 2
        kougo(3, 1) = 3
        kougo(4, 1) = 4
        kougo(1, 2) = -1
        kougo(2, 2) = -2
        kougo(3, 2) = -3
        kougo(4, 2) = -4
    Case "3×3"
        Select Case Right(SpecKougo(KougoNr), 1)
        Case "U"
            kougo(1, 1) = 1
            kougo(1, 2) = 2
            kougo(1, 3) = 3
            kougo(2, 1) = 2
            kougo(2, 2) = 3
            kougo(2, 3) = 1
            kougo(3, 1) = 3
            kougo(3, 2) = 1
            kougo(3, 3) = 2
        Case "V"
            kougo(1, 1) = 3
            kougo(1, 2) = 2
            kougo(1, 3) = 1
            kougo(2, 1) = 1
            kougo(2, 2) = 3
            kougo(2, 3) = 2
            kougo(3, 1) = 2
            kougo(3, 2) = 1
            kougo(3, 3) = 3
        Case Else
            MsgBox "交互作用名末尾でUまたはVが欠落しています"
            End
        End Select
    Case "2×4"
        If Right(SpecKougo(KougoNr), 1) <> "W" Then
            MsgBox "KougoName の末尾""W""をつけなさい"
            End
        End If
        kougo(1, 1) = 1
        kougo(2, 1) = -1
        kougo(1, 2) = 2
        kougo(2, 2) = -2
        kougo(1, 3) = 3
        kougo(2, 3) = -3
        kougo(1, 4) = 4
        kougo(2, 4) = -4
    Case Else
        MsgBox "Fatal error in OrthoTabKougo() No.1"
        End
    End Select
Return

End Sub

Private Function ChangeUtoV(ByVal U As String)
'純粋LN進数(0,1,2,...,(n-1))を直交表表記(1,2,3,...,n)に変換します
'LN<=9の場合のみ有効です
    Dim A(100) As Integer
    Dim keta As Integer, i As Integer
```

```
    keta = Len(U)
    V = ""
    For i = 1 To keta
        A(i) = Val(Mid(U, i, 1)) + 1
        V = V & LTrim(Str(A(i)))
    Next i
    ChangeUtoV = V
End Function

Function ChangeVtoU(ByVal V As String) As String
'直交表表記(1,2,3,...,n)を純粋LN進数(0,1,2,...,(n-1))に変換します
'LN<=9の場合のみ有効です
    Dim A(100) As Integer
    Dim keta As Integer, i As Integer

    keta = Len(V)
    U = ""
    For i = 1 To keta
        A(i) = Val(Mid(V, i, 1)) - 1
        U = U & LTrim(Str(A(i)))
    Next i
    ChangeVtoU = U
End Function

Function Change10toLN(ByVal x As Long, LN As Integer) As String
'10進数xをLN進数に変換して、keta桁文字列(0,1,2)として返します
    Dim t As Long, r As String, LNN As Integer
    Dim keta As Integer, i As Integer, Row As Integer, Col As Integer

    If x = 0 Then Change10toLN = c0: Exit Function
    '【注】探索3行3列のとき9桁
    t = 1&
    r = ""
    If LN <> 0 Then
        Do While (t <= x)
            r = CStr((x Mod (t * LN)) ¥ t) & r
            t = t * LN
        Loop
        keta = Len(r)
    Else
        For Row = SearchRowEnd To SearchRowStart Step -1
            For Col = SearchColEnd To SearchColStart Step -1
                LNN = LevelNum(Col) 'Colは既に与えられている
                r = CStr((x Mod (t * LNN)) ¥ t) & r
                t = t * LNN
            Next Col
        Next Row
    End If
    keta = Len(r)
    If keta < Len(c0) Then
        For i = keta + 1 To Len(c0)
            r = "0" & r
        Next i
    End If
    Change10toLN = r
End Function

Sub InitialSet(cntUstart As Long, cntUstop As Long)
'直交表主効果部読込み
```

```
Dim i As Integer, j As Integer, k As Integer
Dim SearchCellNum As Long, Row As Integer, Col As Integer

SheetName = ActiveSheet.Name
'交互作用指定
Cells(1, 5).Value = "KougoNum ="
Cells(2, 5).Value = "KougoName ="
If Cells(1, 7).Value = "" Then
    MsgBox "KougoNum が指定されていません"
    End
End If
KougoNum = Val(Cells(1, 7).Value)
For i = 1 To KougoNum
    SpecKougo(i) = Cells(2, 6 + i).Value
    If SpecKougo(i) = "" Then
        MsgBox "KougoName が指定されていません"
        End
    End If
Next i
OrthoTitleTop = GetTitleRow("◆直交表", 1) '←Excel 列番号
'【注】直交表タイトルは"◆直交表"の行の第 3 列で設定される
rowStartXL = OrthoTitleTop + 3
colStartXL = 2 '←Excel 列番号
'直交表分岐
OrthoTabName = Cells(OrthoTitleTop, 3).Value
Select Case OrthoTabName
Case "L4+0"
    OrthoRowNum = 4 '直交表行数
    FacterNum = 3 '因子数
Case "L4+1"
    OrthoRowNum = 5 '直交表行数
    FacterNum = 3 '因子数
Case "L4+2"
    OrthoRowNum = 6 '直交表行数
    FacterNum = 3 '因子数
Case "L4+3"
    OrthoRowNum = 7 '直交表行数
    FacterNum = 3 '因子数
Case "L4+4"
    OrthoRowNum = 8 '直交表行数
    FacterNum = 3 '因子数
Case "L6(2x3)" '全組合せ実験
    AllCombExp = True
    OrthoRowNum = 6 '直交表行数
    FacterNum = 2 '因子数
Case "L8(2x4)"
    AllCombExp = True
    OrthoRowNum = 8 '直交表行数
    FacterNum = 2 '因子数
Case "L9(3x3)"
    AllCombExp = True
    OrthoRowNum = 9 '直交表行数
    FacterNum = 2 '因子数
Case "L8+0"
    OrthoRowNum = 8 '直交表行数
    FacterNum = 7 '因子数
Case "L8+1"
    OrthoRowNum = 9 '直交表行数
    FacterNum = 7 '因子数
Case "L8+2"
```

```
    OrthoRowNum = 10 '直交表行数
    FacterNum = 7 '因子数
Case "L8+3(2x4)"
    OrthoRowNum = 11 '直交表行数
    FacterNum = 5 '因子数
Case "L8+4(2x4)"
    OrthoRowNum = 12 '直交表行数
    FacterNum = 5 '因子数
Case "L8+21"
    OrthoRowNum = 29 '直交表行数
    FacterNum = 7 '因子数
Case "L9+0"
    OrthoRowNum = 9 '直交表行数
    FacterNum = 4 '因子数
Case "L9+2"
    OrthoRowNum = 11 '直交表行数
    FacterNum = 4 '因子数
Case "L9+4"
    OrthoRowNum = 13 '直交表行数
    FacterNum = 4 '因子数
Case "L9+6"
    OrthoRowNum = 15 '直交表行数
    FacterNum = 4 '因子数
Case "L18Whole" 'for display of whole Kougo columns
    OrthoRowNum = 18 '直交表行数
    FacterNum = 8 '因子数
Case "L18+0r" '追加行なしで交互作用列１個
    OrthoRowNum = 18 '直交表行数
    FacterNum = 8 '因子数
Case "L18+0s" '因子Ｈ列を削り代わりに交互作用１個を充てる
    OrthoRowNum = 18 '直交表行数
    FacterNum = 7 '因子数
Case "L18+0"
    OrthoRowNum = 18 '直交表行数
    FacterNum = 8 '因子数
Case "L18+1"
    OrthoRowNum = 19 '直交表行数
    FacterNum = 8 '因子数
Case "L18+2"
    OrthoRowNum = 20 '直交表行数
    FacterNum = 8 '因子数
Case "L18+3"
    OrthoRowNum = 21 '直交表行数
    FacterNum = 8 '因子数
Case "L18+4"
    OrthoRowNum = 22 '直交表行数
    FacterNum = 8 '因子数
Case "L18+6"
    OrthoRowNum = 24 '直交表行数
    FacterNum = 8 '因子数
Case "L18+8"
    OrthoRowNum = 26 '直交表行数
    FacterNum = 8 '因子数
Case "L18+9"
    OrthoRowNum = 27 '直交表行数
    FacterNum = 8 '因子数
Case "L18+18"
    OrthoRowNum = 36 '直交表行数
    FacterNum = 8 '因子数
Case "L18+36"
```

```
    OrthoRowNum = 54 '直交表行数
    FacterNum = 8 '因子数
Case Else
    MsgBox "Fatal error in InitialSet() No.1"
    End
End Select
If FacterNum <= 0 Or OrthoRowNum <= 0 Then
    MsgBox "InitialSet 不備です"
    End
End If
SearchCondTop = rowStartXL + OrthoRowNum + 1 '"◆探索条件"のExcel行
SetMitisuTop = rowStartXL + OrthoRowNum + 10 '"◆未知数"のExcel行
'直交表の右側をクリア
Range(Cells(OrthoTitleTop, FacterNum + KougoNum + colStartXL + 1), _
                Cells(OrthoTitleTop + OrthoRowNum + 3, 255)).Select
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
Selection.Borders(xlEdgeLeft).LineStyle = xlNone
Selection.Borders(xlEdgeTop).LineStyle = xlNone
Selection.Borders(xlEdgeBottom).LineStyle = xlNone
Selection.Borders(xlEdgeRight).LineStyle = xlNone
Selection.Borders(xlInsideVertical).LineStyle = xlNone
Selection.Borders(xlInsideHorizontal).LineStyle = xlNone
Selection.HorizontalAlignment = xlCenter
Selection.ClearContents
'直交表主効果部の読み取り
For k = 1 To OrthoRowNum
    For f = 1 To FacterNum
        OrthoTab(k, f) = _
            Val(Cells(k + rowStartXL - 1, f + colStartXL - 1).Value)
    Next f
Next k
For f = 1 To FacterNum + KougoNum
    LevelNum(f) = Cells(rowStartXL + OrthoRowNum, _
                                            colStartXL + f - 1).Value
Next f
'探索条件と探索範囲
If AllCombExp <> True Then
    '探索条件以下をクリア
    Range(Cells(SearchCondTop + 1, 4), Cells(SearchCondTop + 4, 256)).Select
    Selection.ClearContents
    Selection.HorizontalAlignment = xlCenter
    Range(Cells(SearchCondTop + 5, 1), Cells(2000, 256)).Select
    Selection.ClearContents
    Selection.HorizontalAlignment = xlCenter
    Cells(SearchCondTop, 1).Value = "◆探索条件"
    Cells(SearchCondTop + 1, 1).Value = "SearchRowStart"
    Cells(SearchCondTop + 2, 1).Value = "SearchRowEnd"
    Cells(SearchCondTop + 3, 1).Value = "SearchColStart"
    Cells(SearchCondTop + 4, 1).Value = "SearchColEnd"
    Range(Cells(SearchCondTop + 1, 1), Cells(SearchCondTop + 4, 1)).Select
    Selection.HorizontalAlignment = xlLeft
    Selection.Font.Bold = True
    '探索範囲(直交表行列番号)
    SearchRowStart = Val(Cells(SearchCondTop + 1, 3))
    SearchRowEnd = Val(Cells(SearchCondTop + 2, 3))
    SearchColStart = Val(Cells(SearchCondTop + 3, 3))
    SearchColEnd = Val(Cells(SearchCondTop + 4, 3))
    If SearchRowStart < 1 Then
        SearchRowStart = OrthoRowNum
```

```
        Cells(SearchCondTop + 1, 1).Value = "SearchRowStart"
        Cells(SearchCondTop + 1, 3).Value = SearchRowStart
    End If
    If SearchRowEnd < SearchRowStart Then
        SearchRowEnd = SearchRowStart
        Cells(SearchCondTop + 2, 1).Value = "SearchRowEnd"
        Cells(SearchCondTop + 2, 3).Value = SearchRowEnd
    End If
    If SearchColStart < 1 Then
        SearchColStart = FacterNum - 1
        Cells(SearchCondTop + 3, 1).Value = "SearchColStart"
        Cells(SearchCondTop + 3, 3).Value = SearchColStart
    End If
    If SearchColEnd < SearchColStart Then
        SearchColEnd = FacterNum
        Cells(SearchCondTop + 4, 1).Value = "SearchColEnd"
        Cells(SearchCondTop + 4, 3).Value = SearchColEnd
    End If
    If SearchColEnd < SearchColStart Then
        MsgBox "SearchColEnd 値が不適切です"
        End
    End If
    If SearchMode = True Then
        ResultTop = SetMitisuTop + 4 '"◆解リスト"の Excel 行
        ResultLeft = colStartXL
        cutColWidth = SearchColEnd - SearchColStart + 1
        cutRowWidth = SearchRowEnd - SearchRowStart + 1
        '探索範囲表示
        DispSearchArea
        'LN 進数定数セット
        c0 = "": c1 = "": c2 = "": c3 = ""
        For i = 1 To SearchRowEnd - SearchRowStart + 1
            For j = 1 To SearchColEnd - SearchColStart + 1
                c0 = c0 & "0"
                c1 = c1 & "1"
                c2 = c2 & "2"
                c3 = c3 & "3"
            Next j
        Next i
        If SearchNum <= cntUstop Then
            cntUstop = SearchNum - 1 'cntU は 0 からカウント
        End If
        Cells(SearchCondTop + 5, 1).Value = "SearchNum =" & Str(SearchNum)
        Cells(SearchCondTop + 5, 1).HorizontalAlignment = xlLeft
        'スタート&ストップ
        Cells(SearchCondTop + 6, 1).Value = "cntUstart/stop =" _
                                & Str(cntUstart) & " ～" & Str(cntUstop)
        Cells(SearchCondTop + 6, 1).HorizontalAlignment = xlLeft
    End If
Else
    '探索条件以下をクリア
    Range(Cells(SearchCondTop + 1, 1), Cells(SearchCondTop + 4, 256)).Select
    Selection.ClearContents
    Selection.HorizontalAlignment = xlCenter
    Range(Cells(SearchCondTop + 5, 1), Cells(2000, 256)).Select
    Selection.ClearContents
    Selection.HorizontalAlignment = xlCenter
    Cells(SearchCondTop, 1).Value = "◆探索条件：  " _
                & "全組合せ実験(AllCombExp = True)では探索できません"
End If
```

```
End Sub

Sub DispSearchArea()
'Display of Search area
    Dim j As Integer

    If SearchMode <> True Then Exit Sub
    '探索範囲の太線囲み
    Range(Cells(rowStartXL + SearchRowStart - 1, colStartXL + _
            SearchColStart - 1), Cells(rowStartXL + SearchRowEnd - 1, _
            colStartXL + SearchColEnd - 1)).Select
    Selection.Borders(xlDiagonalDown).LineStyle = xlNone
    Selection.Borders(xlDiagonalUp).LineStyle = xlNone
    Selection.Borders(xlEdgeLeft).LineStyle = xlNone
    Selection.Borders(xlEdgeTop).LineStyle = xlNone
    Selection.Borders(xlEdgeBottom).LineStyle = xlNone
    Selection.Borders(xlEdgeRight).LineStyle = xlNone
    Selection.Borders(xlInsideVertical).LineStyle = xlNone
    Selection.Borders(xlInsideHorizontal).LineStyle = xlNone
    Selection.Borders(xlEdgeLeft).Weight = xlMedium
    Selection.Borders(xlEdgeTop).Weight = xlMedium
    Selection.Borders(xlEdgeBottom).Weight = xlMedium
    Selection.Borders(xlEdgeRight).Weight = xlMedium
    '探索範囲に１をセット
    For k = SearchRowStart + rowStartXL - 1 To SearchRowEnd + rowStartXL - 1
        For j = SearchColStart + colStartXL - 1 To SearchColEnd + colStartXL - 1
            Cells(k, j).Value = 1
        Next j
    Next k
End Sub

Sub DispResult(cntP As Long, cntU As Long)
    Dim x As Double, j As Integer

    If cntP = 0 Then Exit Sub
    Cells(ResultTop + cntP + 1, 1).Value = cntP
    Cells(ResultTop + cntP + 1, 2).Value = cntU
    For j = 1 To cutRowWidth
        Cells(ResultTop + cntP + 1, j + 2).Value = V1(j)
    Next j
    For MitisuNr = 1 To MitisuNum
        x = kMatrix(MitisuNr, MitisuNum + 1)
        Cells(ResultTop + cntP + 1, MitisuNr + cutRowWidth + 2).Value = x
    Next MitisuNr
End Sub

Sub DispCalcCondition()
    Dim ZeroEqsNum As Integer
    Dim Margin As Integer

    OrthoEqsNum = OrthoRowNum
    ZeroEqsNum = FacterNum + KougoNum
    Margin = OrthoEqsNum + ZeroEqsNum - MitisuNum
    If Margin < 0 Then
        MsgBox "式数が未知数個数より少ないので計算を打ち切ります。"
        End
    End If
    Cells(SearchCondTop + 1, 12).Value = "OrthoEqsNum =" & Str(OrthoEqsNum)
    Cells(SearchCondTop + 1, 12).HorizontalAlignment = xlLeft
    Cells(SearchCondTop + 2, 12).Value = "ZeroEqsNum =" & Str(ZeroEqsNum)
```

```
    Cells(SearchCondTop + 2, 12).HorizontalAlignment = xlLeft
    Cells(SearchCondTop + 3, 12).Value = "MitisuNum =" & Str(MitisuNum)
    Cells(SearchCondTop + 3, 12).HorizontalAlignment = xlLeft
    Cells(SearchCondTop + 4, 12).Value = "Margin =" & Str(OrthoEqsNum) & "+" & _
                    LTrim(Str(ZeroEqsNum)) & "-" & LTrim(Str(MitisuNum)) & " =" & Str(Margin)
    Cells(SearchCondTop + 4, 12).HorizontalAlignment = xlLeft
    Cells(SearchCondTop + 1, 15).Value = "YkMode = " & YkMode
    Cells(SearchCondTop + 1, 15).HorizontalAlignment = xlLeft
    Cells(SearchCondTop + 2, 15).Value = "MitisuMode = " & MitisuMode
    Cells(SearchCondTop + 2, 15).HorizontalAlignment = xlLeft
End Sub

Sub LESQ_V2(AMN, BM, B, M1 As Integer, N1 As Integer, NSTOP As Integer)
'Least SQuare calculation
    Dim hk As Integer, j As Integer, jm As Integer
    Dim z() As Double, BN() As Double, BX() As Double
    ReDim BX(N1, N1), BN(N1), z(N1)

    For hk = 1 To N1
        For j = 1 To N1
            BX(hk, j) = 0
            For jm = 1 To M1
                BX(hk, j) = BX(hk, j) + AMN(jm, hk) * AMN(jm, j)
            Next jm
        Next j
        z(hk) = 0
        For jm = 1 To M1
            z(hk) = z(hk) + AMN(jm, hk) * BM(jm)
            BN(hk) = z(hk)
        Next jm
    Next hk
    LINSW_V2 BX, BN, N1, NSTOP
    If NSTOP = 0 Then '計算が正常終了
        For hk = 1 To N1
            BM(hk) = BN(hk)
            B(hk) = BN(hk)
        Next hk
    End If
End Sub

Public Sub LINSW_V2(A, AN, N1 As Integer, NSTOP As Integer)
'Linear Equations Sweep-out
    Dim hk As Integer, KK As Integer, M As Integer
    Dim Pvt As Double, swp As Double
    Dim B() As Integer, c() As Double
    ReDim B(N1), c(N1)

    If N1 > MaxN1 Or N1 <= 0 Then
        MsgBox ("方程式の次元数が大きすぎか不正です。")
        Exit Sub
    End If
    For hk = 1 To N1
        B(hk) = hk
    Next hk
    For hk = 1 To N1
        PivotH A, AN, B, N1, hk
        Pvt = A(hk, hk)
        If Pvt < ESC Then
            NSTOP = 9
            NRANK = hk - 1
```

```
            Exit Sub
        Else
            For M = hk To N1
                A(hk, M) = A(hk, M) / Pvt
            Next M
            AN(hk) = AN(hk) / Pvt
            For KK = 1 To N1
                If KK <> hk Then
                    swp = A(KK, hk)
                    For M = hk To N1
                        A(KK, M) = A(KK, M) - swp * A(hk, M)
                    Next M
                    AN(KK) = AN(KK) - swp * AN(hk)
                End If
            Next KK
        End If
    Next hk
    NRANK = hk - 1
    For hk = 1 To N1
        c(hk) = AN(hk)
    Next hk
    For hk = 1 To N1
        AN(B(hk)) = c(hk)
    Next hk
End Sub

Sub PivotH(A, AN, B, N1 As Integer, hk As Integer)
'Pivot calculation
    Dim i As Integer, M As Integer, BB As Integer
    Dim max_a As Double, max_i, max_j As Integer
    Dim c() As Double, ANN As Double
    ReDim c(N1)

    max_a = Abs(A(hk, hk))
    max_i = hk
    For i = hk + 1 To N1
        If Abs(A(i, hk)) > max_a Then
            max_a = Abs(A(i, hk))
            max_i = i
        End If
    Next i
    If max_i <> hk Then
        For M = 1 To N1
            c(M) = A(hk, M)
            A(hk, M) = A(max_i, M)
            A(max_i, M) = c(M)
            ANN = AN(hk)
            AN(hk) = AN(max_i)
            AN(max_i) = ANN
        Next M
    End If
    max_a = Abs(A(hk, hk))
    max_j = hk
    For i = hk + 1 To N1
        If Abs(A(hk, i)) > max_a Then
            max_a = Abs(A(hk, i))
            max_j = i
        End If
    Next i
    If max_j <> hk Then
```

```
        For M = 1 To N1
            c(M) = A(M, hk)
            A(M, hk) = A(M, max_j)
            A(M, max_j) = c(M)
            BB = B(hk)
            B(hk) = B(max_j)
            B(max_j) = BB
        Next M
    End If
End Sub

Sub Keisen(row1 As Integer, col1 As Integer, row2 As Integer, col2 As Integer)
    Range(Cells(row1, col1), Cells(row2, col2)).Select
    Selection.Borders(xlDiagonalDown).LineStyle = xlNone
    Selection.Borders(xlDiagonalUp).LineStyle = xlNone
    Selection.Borders(xlEdgeLeft).LineStyle = xlContinuous
    Selection.Borders(xlEdgeTop).LineStyle = xlContinuous
    Selection.Borders(xlEdgeBottom).LineStyle = xlContinuous
    Selection.Borders(xlEdgeRight).LineStyle = xlContinuous
    If col1 <> col2 Then
        Selection.Borders(xlInsideVertical).LineStyle = xlContinuous
    End If
    Selection.Borders(xlInsideHorizontal).LineStyle = xlContinuous
End Sub

Sub MakeEqs()
'List up of the simultanious linear equations
'連立方程式の式をリストアップします。係数行列表の変数名を用います。
    Dim cnt As Integer, i As Integer, j As Integer, ii As Integer
    Dim Eqs As String

    EqsTop = KeisuMatrixTop + OrthoRowNum + FacterNum + KougoNum
    Cells(EqsTop, 1).Value = "◆連立方程式（実験値式&ゼロ和式）" _
                                    & "  式は係数行列に基づいて作成"
    Cells(EqsTop, 1).HorizontalAlignment = xlLeft
    Cells(EqsTop, 1).Select
    M1 = OrthoRowNum + FacterNum + KougoNum
    N1 = M1
    For k = 1 To M1
        Eqs = "y(" & Cells(KeisuMatrixTop + k - 1, 1).Value & ") = "
        cnt = 0
        For j = 1 To N1
            If kMatrix(k, j) = 1 Then
                cnt = cnt + 1
                If cnt = 1 Then
                    Eqs = Eqs & Cells(KeisuMatrixTop - 1, j + 1).Value
                Else
                    Eqs = Eqs & " + " & Cells(KeisuMatrixTop - 1, j + 1).Value
                End If
            ElseIf kMatrix(k, j) = -1 Then
                cnt = cnt + 1
                If cnt = 1 Then
                    Eqs = Eqs & Cells(KeisuMatrixTop - 1, j + 1).Value
                Else
                    Eqs = Eqs & " - " & Cells(KeisuMatrixTop - 1, j + 1).Value
                End If
            End If
        Next j
        Eqs = Eqs & " = " & Int(Val(y(k) * 1000)) / 1000
        Cells(EqsTop + k, 2).Value = Eqs
```

```
        Cells(EqsTop + k, 2).HorizontalAlignment = xlLeft
    Next k
End Sub

Function GetTitleRow(TitleString As String, TitleCol As Integer)
'タイトル文字列をTitleCol列で探索。見つかれば行番号を返します。
    Dim i As Integer

    For i = 1 To 1000
        If Cells(i, TitleCol).Value = TitleString Then
            GetTitleRow = i
            Exit Function
        End If
    Next i
    MsgBox TitleString & "が見つかりません"
    End
End Function

Sub DispOrthoTab()
'Display the ortogonal table
    Dim k As Integer, j As Integer

    For k = 1 To OrthoRowNum
        For j = 1 To FacterNum + KougoNum
            Cells(rowStartXL + k - 1, colStartXL + j - 1).Value = OrthoTab(k, j)
        Next j
    Next k
End Sub

Function Hantei() As Boolean
'連立方程式式で解あり／なしを判定
    If NSTOP = 0 Then
        If YkMode <> "HAND" Then
            For MitisuNr = 1 To MitisuNum
                If Abs(MitisuVal(MitisuNr) - _
                                kMatrix(MitisuNr, MitisuNum + 1)) > ESC Then
                    Hantei = False
                    Exit Function
                End If
            Next MitisuNr
        End If
        Hantei = True
    Else
        Hantei = False
        Exit Function
    End If
End Function
```