

Complément à l'article « Comptes consolidés : Calculer les pourcentages d'intérêt à l'aide du calcul matriciel et du langage VBA d'EXCEL »

Publié dans la Revue Française de Comptabilité de janvier 2012 :

Explication pas à pas du code source VBA



Benoît RIVIERE

Expert-comptable chez COGEDIAC
& ASSOCIES S.A. à Caen

benoit@auditsi.eu

blog : www.auditsi.eu

Le tableau ci-après explique le fonctionnement du programme VBA. Afin de ne pas rendre trop complexe la compréhension du code source, le programme tel qu'il est reproduit ici est allégé entre autres au niveau des procédures de contrôle ; ainsi, la cohérence des données saisies n'est pas testée (doublons, participations non déclarées...).

Code VBA	Commentaires
<pre>'POURCINT V1.0 08/2011 'Application calculant les pourcentages d'intérêt 'Programmée par Benoît-René RIVIERE, benoit@auditsi.eu Option Explicit '===== 'DECLARATION DES CONSTANTES ET VARIABLES '----- [...]</pre>	
<pre>'---Entités '-----Liste des entités Dim Entités() As String '-----Nombre d'entités Dim NbEntités As Integer '---Détentions '-----Liste des détentions stockée sous forme de tableau à deux dimensions (matrice) Dim MatriceDétentions() As Double 'X, Y : détenteur, détenu Dim MatriceI() As Double Dim MatriceImoinsDétentions() As Double Dim PourcentagesIntérêt() As Double</pre>	
Dim tableau() as type	Définition des variables de type tableau. Les tableaux à deux dimensions ont l'avantage de se présenter sous la même forme que les matrices (tableaux à double entrée). Pour plus d'information sur la déclaration des tableaux, cf Encadré 3. Le type retenu est « double », c'est-à-dire que les données stockées dans les tableaux seront des nombres à décimales.
<pre>'Lecture de la liste des entités Sub LectureEntités() 'Entité 1 = entité fictive 'Entité 2 = entité mère '3, 4... = entités filles [...]</pre>	
<pre>'---Définition de l'entité fictive NbEntités = 1 ReDim Entités(NbEntités) Entités(NbEntités) = "X" '---Lecture des entités Do s = classeurAppli.Sheets(FeuilleEntités).Range(ColonneRéfEntité & NbEntités).Value If s <> "" Then NbEntités = NbEntités + 1 ReDim Preserve Entités(NbEntités) Entités(NbEntités) = s End If Loop While s <> "" 'dès qu'une référence est vierge, la lecture est arrêtée</pre>	

Code VBA	Commentaires
<pre>'Dimensionnement des matrices ReDim MatriceDétentions(1 To NbEntités, 1 To NbEntités) ReDim MatriceI(1 To NbEntités, 1 To NbEntités) ReDim MatriceImoinsDétentions(1 To NbEntités, 1 To NbEntités) ReDim PourcentagesIntérêt(2 To NbEntités) End Sub</pre>	
<pre>Sub LectureEntités() [...]</pre>	<p>Cette procédure lit la liste des entités (Figure 7) et dimensionne les matrices (le redimensionnement des variables tableaux est commenté en Encadré 3).</p>
<pre>Sub ConstructionMatriceDétentions() [...]</pre> <pre>'---Lecture des détentions i = 0 Do i = i + 1 With classeurAppli.Sheets(FeuilleDétentions) s1 = .Range("A" & i).Value s2 = .Range("C" & i).Value détenteur = DonneCodeEntité(s1) détenu = DonneCodeEntité(s2) If (s1 <> "") And (s2 <> "") And (détenteur <> 0) And (détenu <> 0) Then v = .Range("B" & i).Value MatriceDétentions(détenteur, détenu) = v End If End With Loop While (s1 <> "") And (s2 <> "") 'dès qu'une référence est vierge, la lecture est arrêtée 'Traitement de l'entité fictive MatriceDétentions(1, 2) = 1 'la détention de l'entité fictive sur la mère est de 100 % en l'absence de participations de filles sur la mère For détenteur = 2 To NbEntités MatriceDétentions(1, 2) = MatriceDétentions(1, 2) - MatriceDétentions(détenteur, 2) Next détenteur End Sub</pre>	
<pre>Sub ConstructionMatriceDétentions() [...]</pre>	<p>Cette procédure construit la matrice des détentions (reproduite en Figure 2) à partir :</p> <ul style="list-style-type: none"> • du portefeuille-titres (Figure 8) ($\text{MatriceDétentions}(\text{détenteur}, \text{détenu}) = v$) • et de l'entité fictive (cf Encadré 2) ($\text{MatriceDétentions}(1, 2) = 1$). <p>Accès au contenu d'une variable tableau : cf Encadré 4.</p>
<pre>Sub ConstructionMatriceI() [...]</pre> <pre>For i = 1 To NbEntités MatriceI(i, i) = 1 Next i End Sub</pre>	
<pre>Sub ConstructionMatriceI() [...]</pre>	<p>Cette procédure construit la matrice Identité (cf Figure 3) qui ne comprend que des 1 dans sa diagonale : $\text{MatriceI}(i, i) = 1$</p>
<pre>Sub ConstructionMatriceImoinsDétentions() [...]</pre> <pre>For détenteur = 1 To NbEntités For détenu = 1 To NbEntités MatriceImoinsDétentions(détenteur, détenu) = MatriceI(détenteur, détenu) - MatriceDétentions(détenteur, détenu) Next détenu Next détenteur End Sub</pre>	
<pre>Sub ConstructionMatriceImoinsDétentions() [...]</pre>	<p>Cette procédure effectue la soustraction entre les matrices Identité et Détentions (cf Figure 4) : $\text{MatriceImoinsDétentions}(\text{détenteur}, \text{détenu}) = \text{MatriceI}(\text{détenteur}, \text{détenu}) - \text{MatriceDétentions}(\text{détenteur}, \text{détenu})$</p>
<pre>Sub AfficheMatrice() [...]</pre> <pre>classeurAppli.Sheets.Add after:=Worksheets(Worksheets.Count) With classeurAppli.Sheets(classeurAppli.Sheets.Count) .Cells(1, 1).Value = "Matrice I-M" For détenteur = 1 To NbEntités For détenu = 1 To NbEntités</pre>	

Code VBA	Commentaires
<pre> If détenteur = 1 Then .Cells(1, détenu + 1).Value = Entités(détenu) .Cells(3 + NbEntités, détenu + 1).Value = Entités(détenu) End If If détenu = 1 Then .Cells(détenteur + 1, 1).Value = Entités(détenteur) .Cells(détenteur + NbEntités + 3, 1).Value = Entités(détenteur) End If .Cells(détenteur + 1, détenu + 1).Value = MatriceImoinsDétentions(détenteur, détenu) Next détenu Next détenteur .Cells(3 + NbEntités, 1).Value = "Matrice (I-M)^-1" .Range("B" & 4 + NbEntités & ":" & Chr(64 + NbEntités + 1) & NbEntités * 2 + 3).FormulaArray = "=minverse(B2:" & Chr(64 + NbEntités + 1) & NbEntités + 1 & ")" 'FormulaArray équivaut à Ctrl+Maj+Entrée For détenu = 2 To NbEntités PourcentagesIntérêt(détenu) = Cells(4 + NbEntités, détenu + 1) Next détenu End With End Sub </pre>	
<pre> Sub AfficheMatrice () [...]</pre> <p>End Sub</p>	<p>Cette procédure affiche les matrices I-M (Figure 4) et $(I-M)^{-1}$ (Figure 5).</p> <p>classeurAppli.Sheets.Add after:=Worksheets(Worksheets.Count) : ajoute une feuille de calcul (Add) après (After) la dernière la plus à droite dans le classeur.</p> <p>If détenteur = 1 Then [...] End If, If détenu = 1 Then [...] End If : ces deux tests conditionnels assurent l’affichage des entêtes de ligne et de colonne (références des entités) lorsque l’entité en cours de traitement est la première de la liste (n° 1).</p> <p>.Cells(détenteur + 1, détenu + 1).Value = MatriceImoinsDétentions(détenteur, détenu) : affiche chaque élément de la matrice MatriceImoinsDétentions dans les cellules (.Cells) de la feuille de calcul. Le déplacement dans la matrice s’effectue à l’aide de la boucle For... To... Next.</p> <p>.Range("B" & 4 + NbEntités & ":" & Chr(64 + NbEntités + 1) & NbEntités * 2 + 3).FormulaArray = "=minverse(B2:" & Chr(64 + NbEntités + 1) & NbEntités + 1 & ")"</p> <p>=minverse() : fonction Excel qui calcule la matrice inverse (⁻¹).</p> <p>FormulaArray: les calculs matriciels utilisant des fonctions EXCEL nécessite cette fonction en VBA (équivaut à Ctrl+Maj+Entrée pour un calcul matriciel validé manuellement dans une feuille de calcul).</p> <p>Chr(65) : renvoie la lettre A (la lettre A est codifiée 65, B 66... dans la table ASCII).</p>
<pre> Sub AffichePourcentagesIntérêt () Dim détenu As Integer With classeurAppli.Sheets(classeurAppli.Sheets.Count) .Cells(NbEntités * 2 + 1 + 4, 1).Value = "% intérêt" For détenu = 2 To NbEntités .Cells(NbEntités * 2 + détenu + 4, 1).Value = Entités(détenu) With .Cells(NbEntités * 2 + détenu + 4, 2) .NumberFormat = "##.00%" .Value = PourcentagesIntérêt(détenu) End With Next détenu End With End Sub </pre>	
<pre> Sub AffichePourcentagesIntérêt() [...]</pre> <p>End Sub</p>	<p>Cette procédure affiche les pourcentages d’intérêt reproduits en figure 6.</p> <p>.NumberFormat = "##.00%": affiche les nombres sous forme de pourcentages.</p> <p>.Value = PourcentagesIntérêt(détenu) : reprend le pourcentage d’intérêt du groupe sur l’entité (référéncée par la variable détenu) stocké dans la variable tableau PourcentagesIntérêt()</p>

Code VBA	Commentaires
<pre>Sub Calculs() '---Déclaration du classeur de l'application Set classeurAppli = Workbooks(NomClasseurAppli) LectureEntités ConstructionMatriceDétentions ConstructionMatriceI ConstructionMatriceImoinsDétentions AfficheMatrice AffichePourcentagesIntérêt End Sub</pre>	
<pre>Sub Calculs() [...]</pre>	<p>Cette procédure lance les différentes étapes de collecte des données (nom des entités, portefeuille de participations), de calcul des matrices et d’affichage des résultats. Le lancement des différentes étapes de calcul s’effectue en appelant les procédures les unes après les autres : « LectureEntités », « ConstructionMatriceDétentions »...</p>

Les figures mentionnées ci-avant renvoient à l’article publié dans la Revue Française de Comptabilité.

Encadré 3 : Déclaration des variables tableau

Dim tableau(10) as String : la variable tableau aura une taille de 10 emplacements indexés de 0 à 9.

A noter : La déclaration « Option Base 1 » en tête de programme à pour incidence de faire débiter l’index à un au lieu de zéro.

Dim tableau(2 to 10) as String : l’index aura pour intervalle : 2..10.

Dim tableau() as String : la taille de l’index n’est pas fixé et sera défini dans le corps du programme (le tableau est alors redimensionnable). C’est cette dernière méthode de déclaration qui est utilisée dans le programme. Elle permet de déclarer la variable alors même que l’on ignore, à ce stade, le nombre de données à déclarer (en l’occurrence, le nombre d’entités à consolider). Lorsque le nombre de données est connu, on le déclare ainsi : ReDim Entités(Nb). Cette déclaration a pour effet d’effacer toutes les données stockées dans le tableau. Pour redimensionner le tableau sans l’effacer, il est nécessaire d’ajouter le l’instruction « Preserve » à la déclaration : ReDim Preserve Entités(NbEntités).

Encadré 4 : Accès au contenu des variables tableau

Les données contenues dans les variables tableaux à deux dimensions sont organisées de manière comparable aux matrices. Ainsi, une matrice est constituée de lignes (X) et de colonnes (Y). L’intersection de chaque ligne et colonne correspond à une information (comme pour une feuille de calcul Excel) : tableau(x,y)=donnée.

Pour prendre l’exemple de la matrice des détentions (Figure 2), MatriceDétentions(2,1) renvoie 0,02 c’est-à-dire F1 détient 2 % de SM.

ⁱ ASCII : American Standard Code for Information Interchange, standard de codification des lettres, chiffres et autres caractères en informatique.