

# La fonction matricielle IDENTITE()

Jacques VAILLÉ

[jacques.vaille@free.fr](mailto:jacques.vaille@free.fr)

---

## Utilisation de la fonction :

Rappel : comme toute fonction matricielle, dans une feuille Excel, elle doit être validée en appuyant simultanément sur les touches **CTRL + MAJ + ENTREE**.

### **Première forme IDENTITE() :**

Entrée dans une plage carrée de cellules de la feuille, elle la remplit de la matrice identité dont la taille est celle de la plage.

### **Deuxième forme IDENTITE( nb ) :**

Où **nb** est un entier positif. On obtient la matrice identité de taille **nb**.

### **Troisième forme IDENTITE( Matrice ) :**

Où **Matrice** est le nom d'une matrice ou la référence à une plage de cellules (plage carrée bien entendu !). La dimension du résultat est alors celle de **Matrice**.

**Dans chacun des cas, la fonction peut être utilisée à l'intérieur d'une formule.**

## Installer la macro :

On peut choisir une des deux solutions suivantes :

### **Macro complémentaire Identite.xla :**

Ouvrir le classeur **Identite.xls**. Choisissez alors *Fichier>Enregistrer sous* et dans la boîte de dialogue Type : Macro complémentaire. Cela fait, fermez ce classeur.

Fermez Excel puis rouvrez-le. Allez dans le menu *Outils>Macros complémentaire* et cochez la case **Identite**.

### **Classeur des macros personnelles Perso.xls :**

**Ouvrir** le classeur **Identite.xls**. Ouvrez l'**éditeur** : menu *Outils>Macros>Visual Basic Editor*.

Dans la fenêtre des projets, développez **MatIdentite (Identite.xls)**.

Vous devez voir le module **Ident** sous les Modules. En maintenant le bouton de la souris enfoncé, **faites glisser** ce module sur **VBAProject (Perso.xls)**.

Si vous obtenez une erreur : **#NOM?** Faites précéder le nom de la fonction du nom du module : **=Ident.IDENTITE()**.

## Exemples :

### **Première forme dans une formule :**

Matrice de la projection orthogonale sur l'orthogonal du vecteur unitaire « unit » :

fx {=PRODUITMAT(unit;TRANSPOSE(unit))-identite()}						
	3	4	5	6	7	8
						unit
	-0.92857143	0.21428571	-0.14285714		1	0.267261242
	0.21428571	-0.35714286	-0.42857143		3	0.801783726
	-0.14285714	-0.42857143	-0.71428571		-2	-0.534522484

### **La troisième forme**

permet de s'adapter à la taille de la matrice utilisée comme dans l'exemple suivant où la formule permet, à l'aide du solveur de trouver une valeur propre de la matrice :

La cellule variable est appelée lambda (au-dessous de ce terme) on demande au solveur de rendre nulle la valeur de la cellule sélectionnée.

fx {=DETERMAT(matrice-lambda*identite(matrice))}						
	3	4	5	6	7	
	lambda					
	3.6939948		1	2	-2	
			3	1	1	
	2.0661E-07		-1	0	1	

Cela fonctionne quelle que soit la taille de la région à laquelle on a donné le nom de « matrice ».

Cette fonction peut être utilisée comme argument pour une autre fonction :

fx {=INVERSEMAT(2*identite())}			
	3	4	5
	0.5	0	0
	0	0.5	0
	0	0	0.5