

### Algorithme Gauss\_Seidel

Dim a[n,n], b[n], x1[n], x2[n], Er[n]	{ Dimensionnement }
Var	{ Declaration des variables }
i, j, n : Entiers;	
e, S1, S2, Maxer, R : Réels;	
Te : Booléenne	
Début	
Pour i←1,n	{ Boucle de lecture des données }
Pour j←1,n	
Lire a[i,j] ;	{ Matrice a[i,j] du système }
Fin j	
Lire b[i]	{ Vecteur b[i] du système }
Lire x1[i]	{ Vecteur x1[i] : solution initiale proposée }
Fin i	
Te ← Faux	
Tant que Te = Faux faire	{ Boucle d'itérations }
Pour i←1,n	
S1←0	
S2←0	
Pour j←j+1, n	{ Boucle pour sommer les éléments }
S2←S2-a[i,j]*x1[j]	{ a[i,j]*x[j] à l'étape précédente (k) }
Fin j	
Pour j←0,i-1	{ Boucle pour sommer les éléments }
S1←S1-a[i,j]*x2[j]	{ a[i,j]*x[j] à l'étape suivante (k+1) }
Fin j	
x2[i] ←(b[i]+S1+S2)/a[i,i]	{ Calcul de x[i] à l'étape suivante (k+1) }
Er[i] ←abs(x2[i]-x1[i])	{ Calcul de l'erreur Er[i] :    x[i](k+1)- x[i](k)    }
Fin i	
Maxer←Er[i]	{ Recherche de l'erreur maximale dans le vecteur }
Pour i←2,n	{ Erreur Er[i] }
Si Er[i]>Maxer Alors	
Maxer←Er[i]	
Fin Si	
Fin i	
Si Maxer<=e Alors	{ comparaison entre les erreurs : trouvée et }
Te←Vraie	{ autorisée }
Si non	
Pour i←1,n	{ Transfert des valeurs du vecteur solution }
x1[i] ←x2[i]	{ suivante (k+1) au vecteur solution précédente }
Fin Si	
Fin Tant que	
Pour i←1,n	{ Affichage et impression des résultats du vecteur }
Afficher ('x[' ,i,'] =' ,x2[i])	{ solution finale x[i] (k+1) }
Fin i	
Fin	