

## Algorithme Gauss\_Seidel

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